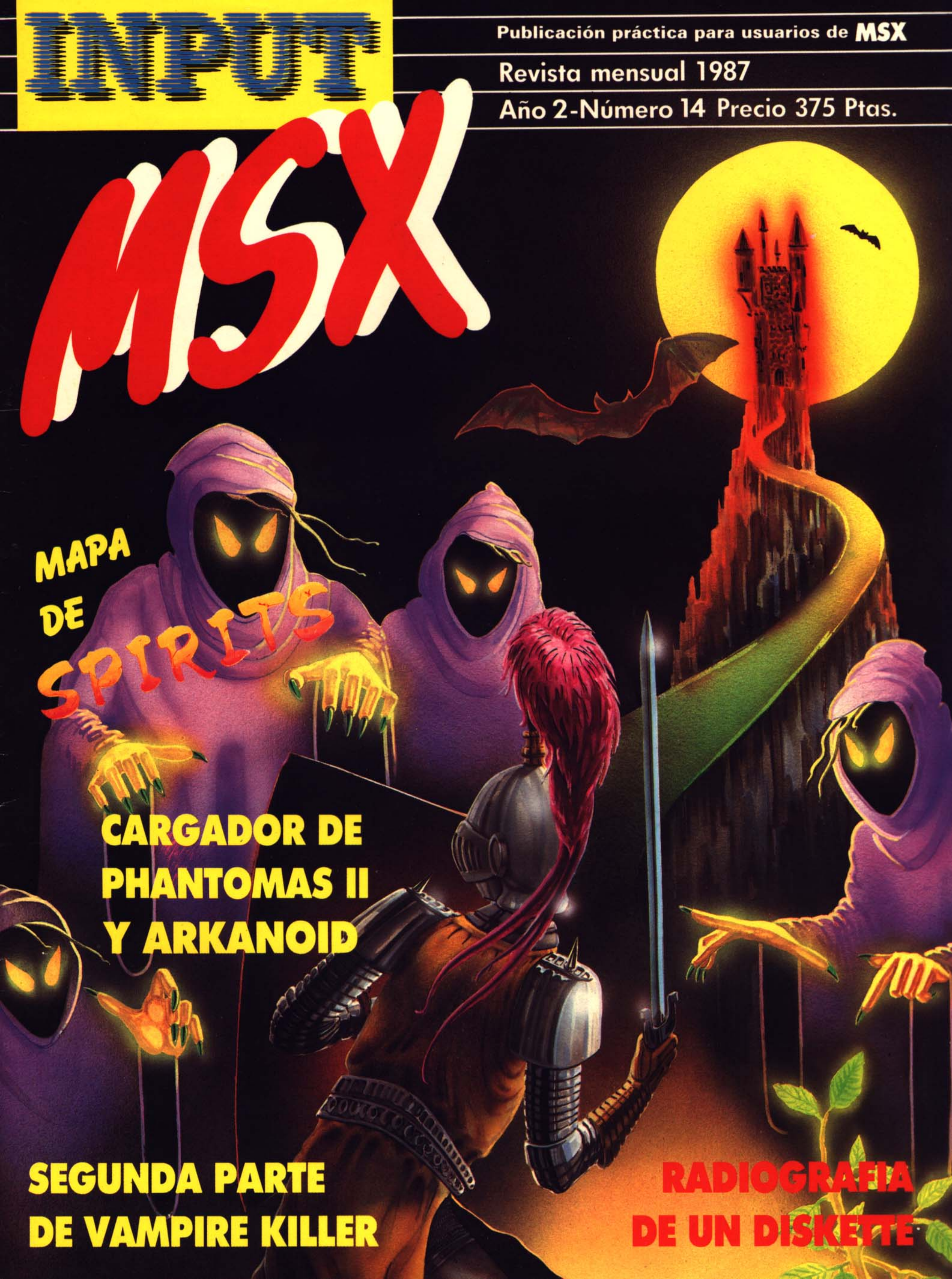

INPUT
Publicación práctica para usuarios de MSX
Revista mensual 1987
Año 2-Número 14 Precio 375 Ptas.
MSX
MAPA DE SPIRITS
CARGADOR DE PHANTOMAS II Y ARKANOID
SEGUNDA PARTE DE VAMPIRE KILLER
RADIOGRAFIA DE UN DISKETTE



RECORTA Y ENVIA ESTE CUPON A KONAMI SHOP.
FRANCISCO NAVACERRADA, 19. 28028 MADRID.

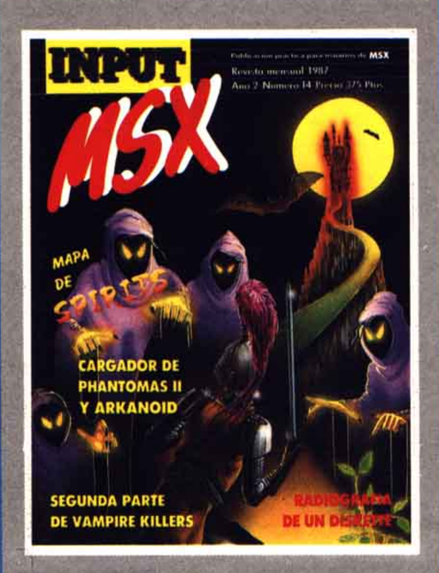
TITULO: ______

NOMBRE Y APELLIDOS: ______

DIRECCION: ______ COD. POSTAL: ______

POBLACION: ______ PROVINCIA: ______

FORMA DE PAGO: CONTRARREMBOLSO ☐ POR TALON BANCARIO ☐



AÑO 2 NUMERO 14

DIRECTOR: Manuel Pérez
DIRECTOR DE ARTE: Luis F. Balaguer
REALIZACION GRAFICA: Didac Tudela
COLABORADORES: Antonio Piego, Xavier Ferrer, Julio García, Ramón Rabaso, Javier de la Fuente, Equipo Molisoft, Jaime Mardones, Irene Alcaraz, Enric Abad, Carles Bartolomé, Angels Alvarez
FOTOGRAFIA: Joan Boada, Ernesto Walfisch

INPUT MSX es una publicación de PLANETA-DE AGOSTINI, S.A.

GERENTE DIVISION DE REVISTAS:
Sebastián Martínez

PUBLICIDAD: INTERMEDIA, S.A.
Gral. Moscardó, 5 - 3.° A
28020 MADRID
Teléf. (91) 442 70 44

FOTOMECANICA: TECFA, S.A.

IMPRESION: Sirven Grafic
c./Gran Vía, 754-756. 08013 Barcelona
Depósito legal: B. 38.115-1986

SUSCRIPCIONES: EDISA
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Aribau, 185, 1.°
08021 Barcelona
DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Calle B, n.° 11. Sector B. Zona Franca
08004 Barcelona

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

INPUT MSX es independiente y no está vinculada a los distribuidores del estándar.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.



SUMARIO



VACACIONES

Se acercan las vacaciones de verano y las noches estivales se prevén largas, cálidas y fantasiosas. Los ordenadores serán el centro de atención de todas las noctámbulas miradas, convirtiendo los aburridos mitines/veladas en fiestas multicolores de sonidos, música y explosiones.

Poco a poco, los joysticks ocupan sus puertos/puestos, el cansancio se va adueñando de nuestras muñecas y los reflejos decaen al paso del tiempo. Llega el final y las despedidas, regresando el silencio a la paz del hogar, a la pantalla del monitor.

El ordenador, compañero de desventuras alienígenas, descansa de sus batallas galácticas interestelares. Las luces se apagan y una nueva etapa comienza, más intimista, más satisfactoria y provechosa: llegar al corazón del microprocesador, seducir al programa residente en memoria, descubrir sus interioridades... Eso es lo que podríamos llamar seducción de un programa.

La noche es un buen estadio para conseguir records, olímpicos o mentales («mens sana in corpore sano»), un circuito donde correr contra reloj contra otros programadores, un «campus» de soledad y duelo entre la máquina y el hombre.

En esas horas, el BASIC se convierte en delator y el código máquina en enemigo. Las sentencias chocan contra los errores sintácticos y los desensambladores son paladines de causas aun por defender. La lucha es ardua, cruel, salvaje,... los párpados sucumben al sueño, la mente obnubilada es presa de ciegas fantasías y lo que creíamos poder hacer se convierte en utopía...

Una nueva mañana clarea en el horizonte y el RETURN del cansancio nos RESETEA hacia otra SUBRUTINA: la del sueño.

RADIOGRAFIA DE UN DISKETTE

A la mayoría de usuarios de diskette les habrá ocurrido en alguna ocasión lo siguiente: insertar un disco repleto de programas, teclear files y ver que el ordenador nos daba un mensaje, terrorífico augurio, de error: «Disk I/O Error».

Contra ello no podíamos hacer nada, toda la información contenida en el diskette estaba irremediablemente perdida. La caída, el mal uso del material magnético, la humedad, el mal conocimiento de algunos comandos (entre ellos «kill», «copy», «format» en disk-BASIC), la no protección contra escritura, etc... hacían peligrar la «vida» interior del disco de 5 1/4 o de 3 1/2.

Contra este tipo de «enfermedades patológicas» al almacenamiento de información, os presentamos dos programas que serán muy útiles a aquellos que les guste «investigar» dentro de los procesos de «copia y verificación»:

El primero es un «copión» de sectores para discos de una cara que os permitirá hacer copias de seguridad, indispensables para usuarios con alto riesgo de pérdida (por ejemplo, los que tengan unidad de disco de 5 1/4).

```
100 SCREEN0:CLEAR50,34978!:CLS:KEYOFF:WIDTH40
105 DEFINTA-Z
110 LOCATE10,5:PRINT"COPIADOR DE SECTORES"
115 GOSUB260
120 LOCATE4,15:BEEP:PRINT"POSEES 1 O 2 UNIDADES DE DISCO:";
125 X$=INKEY$:IFX$=""THEN125
130 X=VAL(X$):IFX<1ORX>2THEN120
135 LOCATE8,20:PRINT"UNIDADES: ";X;SPC(20)
140 FOR F=0TO719 STEP 48
145 IFX=2THEN165
150 FORA=1TO9999:NEXT:BEEP:BEEP
155 LOCATE0,20:BEEP:PRINT"INSERTAR EL DISCO ORIGEN Y PULSAR RETURN"
160 X$=INKEY$:IFX$=CHR$(13)THEN 165 ELSE 160
165 POKE 34992!,0
170 POKE 34986!,INT (F/256)
175 POKE 34985!,FMOD 256
180 POKE 34990!,175
185 LOCATE0,8:PRINT"LECTURA SECTOR: ";F;"-";F+48
190 A=USR(0)
195 GOSUB 320
200 IFX=2THEN 220
205 FORA=1TO9999:NEXT:BEEP:BEEP
210 LOCATE0,20:BEEP:PRINT"INSERTAR EL DISCO DESTINO Y PULSAR RETURN"
215 X$=INKEY$:IFX$=CHR$(13)THEN 220 ELSE 215
220 POKE 34992!,0
225 IFX=2THENPOKE 34992!,1
230 POKE 34990!,55
235 LOCATE0,8:PRINT"ESCRITURA SECTOR: ";F;"-";F+48
240 A=USR(0)
245 GOSUB320
250 NEXT F
255 LOCATE0,20:BEEP:PRINT"COPIA TERMINADA...":END
```

```
260 FORI=0TO19:READ A
265 POKE34980!+I,A:NEXT
270 DEFUSR=34980!:RETURN
275 DATA 6,48
280 DATA14,248
285 DATA17,0,0
290 DATA33,184,136
295 DATA175
300 DATA62,1
305 DATA205,68,1
310 DATA50,163,136
315 DATA201
320 IFPEEK(34979!)<>0THENPRINT"ERROR EN LA OPERACION":STOP
325 RETURN
```

El segundo programa es la «pantalla» que nos permitirá verificar el estado de nuestros diskettes, ver cómo se hallan de salud y determinar las causas de una posible y parasitaria enfermedad.

¿Qué información se guarda en el diskette? ¿Dónde se localiza? ¿Qué formato tiene?

Estas y otras preguntas nos hacemos cuando queremos investigar dentro del diskette. En este artículo vamos a tratar de aclarar algunas dudas al respecto, viendo qué tipo de información se guarda, dónde y cómo localizarla.

En el diskette, la información se almacena en círculos concéntricos llamados PISTAS, dependiendo del tipo de disco tendremos 40 u 80 pistas, en una o dos caras; desde el punto de vista lógico, cada pista se divide a su vez en 9 porciones llamadas SECTORES y cada sector tiene una longitud de 512 bytes. Físicamente, la unidad de diskette lee o graba otros bloques llamados CLUSTERS que tienen una longitud de dos sectores consecutivos, es decir 1024 bytes; ésta es la longitud mínima que ocupará cualquier fichero.

Cuando el tamaño es mayor el sistema operativo busca en una tabla que se encuentra al comienzo del disco llamada F.A.T. qué clusters están libres, encadenándolos y dejándolos marcados como ocupados, así hasta preparar el espacio necesario para el fichero, después actualiza la primera entrada libre del directorio con el nombre del fichero y otras informaciones y por fin graba los datos.

Vamos a ver un poco más detenidamente todas las áreas que han tomado parte de este proceso.

En un disco se pueden diferenciar cuatro zonas que se reservan e inicializan cuando se formatea el disco.

REGISTRO DE AUTOARRANQUE, llamado BOOT, ocupa el sector lógico cero y es un pequeño programa que verifica que el disco es del formato D.O.S.

La TABLA DE ASIGNACION DE ARCHIVOS (F.A.T.) guarda la información de las cadenas de *clusters* que forman los archivos y de los clusters que están libres; cada entrada ocupa 12 bits (1,5 bytes) y tiene 356 entradas (dependiendo del tipo de disco), está situada en los sectores uno y dos. En los sectores tres y cuatro se guarda otra copia a efectos de validación y seguridad.

A continuación, ocupando siete sectores se encuentra el DIRECTORIO, donde se almacena información de los ficheros como el nombre, el tamaño, fecha de creación, etc.

Por último tenemos el AREA DE DATOS que tiene un total de 720 sectores aproximadamente, también dependiendo del tipo de disco, que es donde se guardan los ficheros y programas.

Veamos ahora las dos áreas fundamentales que utiliza el sistema operativo para gestionar el espacio del disco y localizar los ficheros.

En el sector número cinco empieza el área del directorio; esta área contiene una entrada de 32 posiciones por cada fichero que hay en el disco, hasta un máximo de 112 ficheros.

Cada entrada tiene el formato siguiente:

00-07-Nombre del archivo. La primera posición nos muestra el estado de esta entrada del directorio. &H00 indica fin del directorio, se utiliza para hacer las búsquedas más cortas, &HE5 indica que esta entrada se ha usado pero el archivo ha sido borrado (también se habrán borrado a &H000 las entradas de la F.A.T. correspondientes al fichero), cualquier otro carácter será el primero del nombre de un archivo activo.

08-10-Extensión del nombre del archivo, es decir las tres últimas letras después del punto en el nombre.

11-11-Atributo de archivo. Cualquier carácter distinto de &H00, hará que no aparezca en las búsquedas de

directorio; se puede utilizar para ocultar archivos; en MSX-DOS también se puede indicar en esta posición que el archivo es sólo de lectura o del sistema.

12-21-Reservado, lo utiliza el sistema operativo y no deberá ser modificado.

22-23-Hora de creación o modificación del archivo; en estas dos posiciones la información se guarda así:

```
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
I———hh———I I—mm—-I I——ss——I
```

Siendo hh de 0 a 23, mm de 0 a 59 y ss incrementos de dos segundos.

24-25-Fecha de creación o modificación del archivo teniendo este formato:

```
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
I————aa————I I-mm-I I—dd—-I
```

Siendo aa de 0 a 119, para obtener el año a esta cifra se le suma 1980, mm de 1 a 12 y dd de 1 a 31.

26-27-Cluster de comienzo del archivo.

28-31-Tamaño del archivo en octetos.

Si quisiéramos ver el contenido de un fichero, con esta información seguiríamos los siguientes pasos:

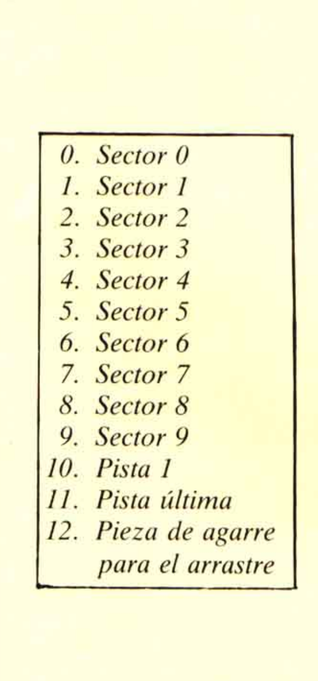
1-Con el cluster de comienzo del fichero se calcula el sector lógico.

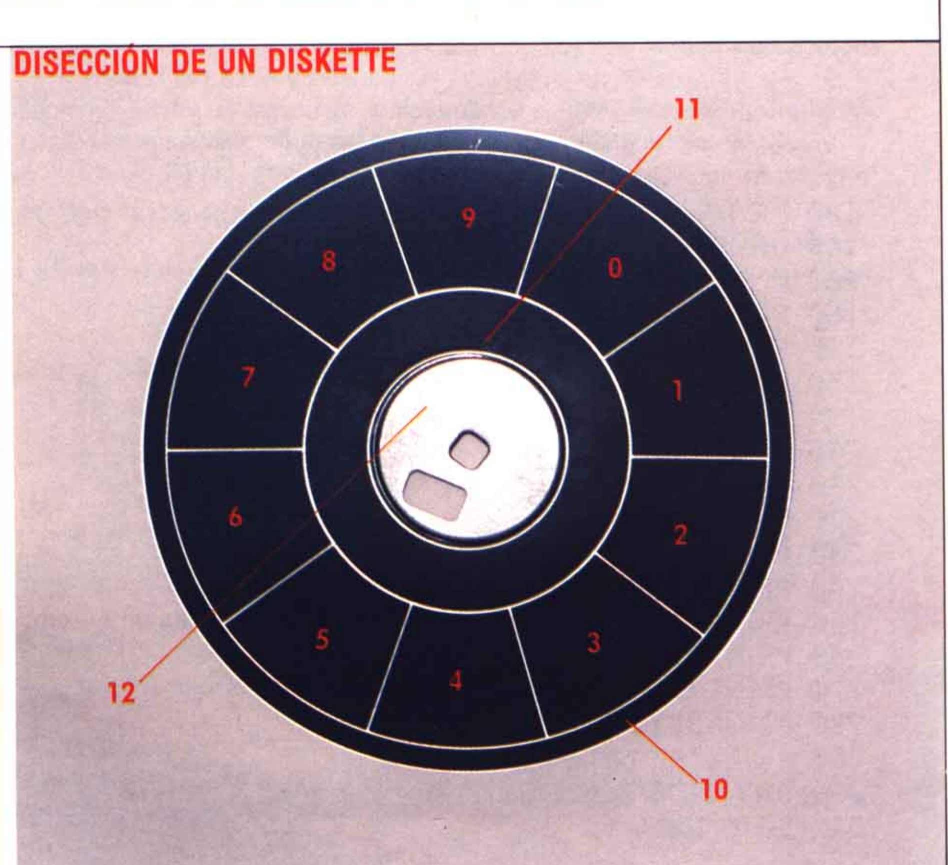
sector=(n.cluster − 2) * 2 + 12

2-Mediante la función A$=DSKI$(0,sector), se lee el sector deseado, que quedará en el buffer o área de lectura de 512 posiciones, que está situado en la dirección indicada por:

DIR=PEEK(&HF351)+256*PEEK(&HF352)

Si el fichero ocupara más de un cluster, tendríamos que recurrir a la F.A.T. y extraer de ahí la cadena de





clusters que están asignados.

En esta tabla, cada entrada (12 bits) representa un cluster del disco y contiene tres caracteres hexadecimales, &H000 si el cluster está libre, &HFF8 o &HFFF indica que éste es el último cluster del archivo, &HFF7 indica que es un cluster defectuoso y cualquier otro carácter nos indicará el siguiente cluster asignado del archivo.

Por razones de rendimiento, se mantiene una copia de esta tabla en memoria, en la dirección dada por:

DIR=(PEEK(&HF351)+256*PEEK(&HF352))+1024

La forma de almacenar esta información, es un tanto peculiar, por lo que un ejemplo aclarará mejor su funcionamiento.

Imaginemos un fichero PROG. BAS que ocupa los clusters 002, 003 y 004; los primeros bytes de la F.A.T. estarán así:

FD FF FF 03 40 00 FF 0F 00 00 00 00

Las dos primeras entradas contienen los caracteres &HFDF &HFFF, que indican el tipo de disco y en la tercera entrada se representa al cluster 002, primero del área de datos.

Ya hemos dicho que cada entrada ocupa 1,5 bytes (tres caracteres hexadecimales) y se forma de la siguiente manera.

```
          FD FF FF 03 40 00 FF 0F 00 00 00 00

          FDF  FFF  003 004  FFF 000  000 000

ENTRADA    0    1    2   3    4   5    6   7
```

La entrada de PROG. BAS en el directorio nos indica que el primer cluster ocupado es el 002 y como vemos la entrada correspondiente a ese cluster indica 003, esto es, el siguiente asignado; en la 003 tenemos 004 y en la entrada correspondiente al cluster 004, &hfff que nos indica que éste es el último de la cadena y el fin del fichero.

En el programa que viene a continuación, se muestran varios ejemplos y se utiliza esta información para obtener listados del directorio, mapas de utilización del disco o un listado de la F.A.T.

```
120 KEYOFF:WIDTH 40:CLEAR3500,&HBFFF:ON STOP GOSUB 1650:STOP ON:DIMDI$(112)
140 ONKEYGOSUB,,,,,,,,,,1590:KEY(10)ON
150 CLS
160 PRINT" ** RADIOGRAFIA DE UN DISKETTE **"
170 PRINT
180 PRINT" RAFAEL LORENZO 2.0"
190 PRINT
200 PRINT:PRINT:PRINT" 1-LISTADO DEL DIRECTORIO"
210 PRINT:PRINT" 2-MAPA DE OCUPACION DEL DISCO"
220 PRINT:PRINT" 3-LISTADO DE LA F.A.T."
230 PRINT:PRINT" 4-LISTAR Y ACTUALIZAR SECTORES"
240 PRINT:PRINT" 5-BUSCAR UN PROGRAMA
250 PRINT:PRINT" 6-FIN DE TRABAJO"
260 PRINT:PRINT" F10-HARDCOPY PANTALLA"
270 PRINT:PRINT:PRINT" SELECCIONA UNA OPCION"
280 A$=INKEY$:IF A$="" THEN 280
290 A=INSTR("123456",A$)
300 ON A GOTO 320,1720,2130,2570,3460,3450
310 GOTO 100
```

```
320 WIDTH40
330 '
340 '
350 '
360 '
370 '
380 '
390 REM ESTE PROGRAMA LISTA EN PANTALLA EL CONTENIDO DE UN DISKETTE INDICANDO
    EL NOMBRE, TIPO DE FICHERO, CLUSTER Y SECTOR DE COMIENZO, SI SE TRATA DE UN
    PROGRAMA EN CODIGO MAQUINA TAMBIEN DA INFORMACION DE LAS DIRECCIONES DE
    COMIENZO, FIN Y EJECUCION
400 REM EN EL CASO DE FICHEROS QUE HAN SIDO BORRADOS (TIPO *DEL*) SOLO DA
    INFORMACION DEL NOMBRE, CLUSTER Y SECTOR DE COMIENZO
410 CLS:IM=0:NP=0:SWFI=0
420 PRINT:PRINT:PRINT:PRINT" * * LISTADO DEL DIRECTORIO * *"
430 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT" QUIERES SALIDA POR IMPRESORA S/N"
440 A$=INKEY$:IF A$=""THENGOTO440
450 IF A$="S" THEN IM=1:LPRINTCHR$(27);"E";ELSE IM=0:REM ESCRIBE COMPRIMIDO
460 PRINT:PRINT:PRINT" COLOCA EL DISCO Y PULSA UNA TECLA"
470 A$=INKEY$:IF A$=""THENGOTO470
480 PRINT:PRINT:PRINT:PRINT" P R O C E S A N D O . . . "
490 DI=PEEK(&HF351)+256*PEEK(&HF352)
500 REM LECTURA DEL DIRECTORIO
510 NP=-1
520 FOR X=5 TO 11
530 A$=DSKI$(0,X)
540 FOR I=DI TO DI+480STEP 32
550 A$=""
560 REM BUSQUEDA DEL NOMBRE
570 IFPEEK(I)=&HE5 THEN GOTO 810
580 IFPEEK(I)=0 THEN X=11:I=DI+480:GOTO 810
590 NP=NP+1
600 FOR J=0 TO 10
610 IF J=8 THEN IF CHR$(PEEK(I+J))<>" " THEN A$=A$+"." ELSE A$=A$+" "
620 A$=A$+CHR$(PEEK(I+J))
630 NEXT J
640 IF LEN(A$) < 12 THEN A$=A$+STRING$((12-LEN(A$))," ")
650 REM CALCULO DEL CLUSTER DE COMIENZO
660 CL=PEEK(I+26)+256*PEEK(I+27)
670 B$="000"+HEX$(CL)
680 B$=RIGHT$(B$,3)
690 A$=A$+B$
700 REM CALCULO DEL SECTOR LOGICO
710 SE=(CL-2)*2+12
720 B$=STR$(SE):B$="000"+RIGHT$(B$,(LEN(B$)-1))
730 B$=RIGHT$(B$,3)
740 A$=A$+B$
750 REM CALCULO DE LA LONGITUD
760 LO=(PEEK(I+28)+256*PEEK(I+29))+256*(PEEK(I+30)+256*PEEK(I+31))
770 B$=STR$(LO):B$="00000"+RIGHT$(B$,(LEN(B$)-1))
```

```
780 B$=RIGHT$(B$,5)
79Ø A$=A$+B$
8ØØ DI$(NP)=A$
81Ø NEXT I
82Ø NEXT X
83Ø P=Ø
84Ø FORM=ØTONP-1
85Ø IF MID$(DI$(M),16,3)>MID$(DI$(M+1),16,3)THEN SWAP DI$(M),DI$(M+1):P=P+1
86Ø NEXTM
87Ø IF P<>ØTHEN GOTO83Ø
88Ø FORN=ØTONP
89Ø A=VAL(MID$(DI$(N),16,3))
9ØØ A$=DSKI$(Ø,A)
91Ø TI=PEEK (DI)
92Ø IFTI=255THENDI$(N)=DI$(N)+"BAS.        ":GOTO1Ø4Ø
93Ø IFTI=49THENDI$(N)=DI$(N)+"ASCI        ":GOTO1Ø4Ø
94Ø IFTI<>254THENDI$(N)=DI$(N)+"DOS         ":GOTO 1Ø4Ø
95Ø A$=HEX$(PEEK(DI+1)+256*PEEK(DI+2))
96Ø IF LEN(A$)<4THEN A$=STRING$(4-LEN(A$),"Ø")+A$
97Ø DI$(N)=DI$(N)+"C.M."+A$
98Ø A$=HEX$(PEEK(DI+3)+256*PEEK(DI+4))
99Ø IF LEN(A$)<4THEN A$=STRING$(4-LEN(A$),"Ø")+A$
1ØØØ DI$(N)=DI$(N)+A$
1Ø1Ø A$=HEX$(PEEK(DI+5)+256*PEEK(DI+6))
1Ø2Ø IF LEN(A$)<4THEN A$=STRING$(4-LEN(A$),"Ø")+A$
1Ø3Ø DI$(N)=DI$(N)+A$
1Ø4Ø NEXT N
1Ø5Ø REM CLASIFICAR ENTRADAS
1Ø6Ø P=Ø
1Ø7Ø FORM=ØTONP-1
1Ø8Ø IF DI$(M)>DI$(M+1)THEN SWAP DI$(M),DI$(M+1):P=P+1
1Ø9Ø NEXTM
11ØØ IF P<>ØTHEN GOTO1Ø6Ø
111Ø REM LISTADO DE LOS RESULTADOS
112Ø GOSUB137Ø:X=4
113Ø FORN=ØTONP
114Ø LI=LI+1:IF LI>19 THENGOSUB 134Ø
115Ø PRINTMID$(DI$(N),1,12); " " ;
116Ø PRINTMID$(DI$(N),24,4);" ";
117Ø PRINTMID$(DI$(N),13,3);"-";
118Ø PRINTMID$(DI$(N),16,3);" ";
119Ø PRINTMID$(DI$(N),28,4);" ";
12ØØ PRINTMID$(DI$(N),32,4);" ";
121Ø PRINTMID$(DI$(N),36,4);
122Ø IF IM=ØTHEN GOTO132Ø
123Ø LPRINTMID$(DI$(N),1,12);" ";
124Ø LPRINTMID$(DI$(N),24,4);" ";
125Ø LPRINTMID$(DI$(N),13,3);"-";
126Ø LPRINTMID$(DI$(N),16,3);" ";
127Ø LO=VAL(MID$(DI$(N),19,5))
```




```
128Ø LPRINTUSING"#####,";LO;
129Ø LPRINT" ";MID$(DI$(N),28,4);" ";
13ØØ LPRINTMID$(DI$(N),32,4);" ";
131Ø LPRINTMID$(DI$(N),36,4);CHR$(11);
132Ø NEXT
133Ø GOTO143Ø
134Ø REM RUTINA DE PANTALLA LLENA
135Ø PRINT:PRINT" **PULSAR TECLA PARA CONTINUAR**"
136Ø IF IM<>1 THEN B$=INKEY$:IF B$="" THEN 136Ø
137Ø CLS:PRINT"ACTIVO TIPO CLT-SEC COM. FIN. EJE.";
138Ø PRINT" ========== ==== ===== ==== ==== ====";
139Ø IF IM=1 AND SWFI=Ø THEN LPRINT"ARCHIVO TIPO CLT-SEC LONG. COM. FIN. JE. ";
     CHR$(1Ø);
14ØØ IF IM=1 AND SWFI=Ø THEN LPRINT "========== ==== ===== =====
     ==== ==== === ";CHR$(1Ø);
141Ø LI=1:SWFI=1
142Ø RETURN
143Ø REM FIN DE DIRECTORIO
144Ø PRINT:PRINT"==========FIN DEL DIRECTORIO======="
145Ø IF IM=1 THEN LPRINT:LPRINT"==========FIN DEL DIRECTORIO========"
146Ø SWFI=1
147Ø GOSUB 135Ø
148Ø CLS:PRINT"ARCHIVOS ARCHIVOS ARCHIVOS ";
149Ø PRINT"========== ========== ==========";
15ØØ FILES:PRINT:PRINT:PRINT USING"####";NP+1;:PRINT" FICHEROS"
151Ø IF IM=1 THEN LPRINT:LPRINT:LPRINT USING"####";NP+1;:LPRINT" FICHEROS"
```

```
1520 PRINT:PRINT USING "####";DSKF(Ø);:PRINT" KBYTES LIBRES"
1530 IF IM=1 THEN LPRINT:LPRINT USING"####";DSKF(Ø);:LPRINT" KBYTES LIBRES"
1540 PRINT"QUIERES LEER OTRO DISKETE S/N"
1550 A$=INKEY$:IF A$="" THEN GOTO 155Ø
1560 IF IM=1 THEN LPRINT CHR$(12)
1570 IF A$="S" OR A$="s" THEN GOTO32Ø
1580 GOTO 1Ø
1590 REM *** HARD COPY SCREEN Ø ***
1600 FOR K=Ø TO 96Ø
1610 A=VPEEK(BASE(Ø)+K)
1620 LPRINT CHR$(A);
1630 IF L=39 THEN LPRINT:L=Ø:ELSE L=L+1
1640 NEXT K
1650 PLAY"CECE"
1660 RETURN
1670 '
1680 '
1690 '
1700 '
1710 '
1720 REM RUTINA DE SITUACION DE CLUSTERS EN EL DISCO
1730 WIDTH 36
1740 CLS:PRINT:PRINT:PRINT:PRINT " * * SITUACION DE CLUSTERS * *"
1750 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT" COLOCA EL DISCO Y PULSA UNA TECLA"
1760 IF INKEY$=""THEN176Ø
1770 CLS:LOCATE1Ø,1Ø:PRINT"PROCESANDO..."
1780 FORO=1TO2
1790 B$=DSKI$(Ø,O)
1800 B=(PEEK(&HF351)+256*PEEK(&HF352))
1810 FORN=Ø TO 511
1820 A=PEEK(B+N):POKE(&HCØØØ+M),A
1830 M=M+1
1840 NEXT
1850 NEXT
1860 CLS:PRINT" * SITUACION DE CLUSTERS OCUPADOS *"
1870 PRINT" ===================="
1880 PRINT:PRINT CHR$(255)" = OCUPADO B = BOOT F = F.A.T.":PRINT:PRINT".=LIBRE D =
     DIREC. * = DEFT."
1890 PRINT:PRINT:PRINT"BFDD";
1900 FOR N= &HCØØØTO&HC3ØØ STEP 3
1910 FOR M=Ø TO 2
1920 A=PEEK(N+M)
1930 A$=HEX$(A):IFLEN(A$)<2THENA$="Ø"+A$
1940 B$=B$+A$
1950 NEXT
1960 C$=MID$(B$,4,1)+MID$(B$,1,2)
1970 IFC$="ØØØ"THENPRINT".";ELSEIFC$="FFØ"THENPRINT"H";
     ELSEIFC$="FF7"THENPRINT"";ELSEPRINTCHR$(255);
1980 C$=MID$(B$,5,2)+MID$(B$,3,1)
1990 IFC$="ØØØ"THENPRINT"".;ELSEIFC$="FFØ"THENPRINT"H";
```

```
     ELSEIFC$="FF7"THENPRINT"";ELSEPRINTCHR$(255);
2ØØØ B$=""
2Ø1Ø X=X+2:IFX=356THEN2Ø3Ø
2Ø2Ø NEXT
2Ø3Ø PRINT:PRINTDSKF(Ø)" CLUSTERS LIBRES
2Ø4Ø PRINT:PRINT"PULSA UNA TECLA PARA MENU"
2Ø5Ø IF INKEY$=""THEN2Ø5Ø
2Ø6Ø GOTO 1Ø
2Ø7Ø '
2Ø8Ø '
2Ø9Ø GOSUB137Ø:X=4
21ØØ '
211Ø '
212Ø '
213Ø REM LISTADO DE LA TABLA DE ASIGNACION DE CLUSTERS
214Ø WIDTH 4Ø
215Ø CLS:LOCATE 6,5:PRINT"** LISTADO DE LA F.A.T. **"
216Ø PRINT:PRINT:PRINT" QUIERES SALIDA POR IMPRESORA S/N"
217Ø A$=INKEY$:IF A$="" THEN 217Ø
218Ø CLS:LOCATE 1Ø,1Ø:PRINT"PROCESANDO..."
219Ø IF A$="S" OR A$="s" THEN IM=1 ELSE IM=Ø
22ØØ FORO=1TO2
221Ø B$=DSKI$(Ø,O)
222Ø B=(PEEK(&HF351)+256*PEEK(&HF352))
223Ø FORN=Ø TO 511
224Ø A=PEEK(B+N):POKE(&HCØØØ+M),A
225Ø M=M+1
226Ø NEXT
227Ø NEXT
228Ø N=&HCØØØ
229Ø CLS:FOR L=1TO 7:PRINT:IF IM=1 THEN LPRINT CHR$(1Ø);CHR$(1Ø);
23ØØ FOR M=CO TO CO+9
231Ø A$=HEX$(M):A$=STRING$(3-LEN(A$),"Ø")+A$
232Ø PRINTA$" ";:IF IM=1 THEN LPRINTA$;" ";
233Ø NEXT
234Ø IF IM=1 THEN LPRINTCHR$(1Ø);
235Ø FOR I=1 TO 5
236Ø FOR M=Ø TO 2
237Ø A=PEEK(N+M)
238Ø A$=HEX$(A):IFLEN(A$)<2THENA$="Ø"+A$
239Ø B$=B$+A$
24ØØ NEXT
241Ø C$=MID$(B$,4,1)+MID$(B$,1,2)
242Ø PRINTC$;" ";:IF IM=1 THEN LPRINTC$;" ";
243Ø C$=MID$(B$,5,2)+MID$(B$,3,1)
244Ø PRINTC$;" ";:IF IM=1 THEN LPRINTC$;" ";
245Ø B$=""
246Ø X=X+2
247Ø N=N+3:IF X=36Ø THEN GOTO 255Ø
248Ø NEXT
```

```
2490 CO=CO+10
2500 NEXT
2510 PRINT:PRINT"PULSA UNA TECLA PARA SEGUIR"
2520 IF IM=0 THEN A$=INKEY$:IF A$=""THEN 2520
2530 GOTO 2290
2540 IF IM=1THEN LPRINTCHR$(12):IM=0
2550 PRINT:PRINT:PRINT"PULSA UNA TECLA PARA MENU"
2560 A$=INKEY$:IF A$=""THEN 2560 ELSE GOTO10
2570 '
2580 '
2590 '
2600 '
2610 '
2620 REM RUTINA DE ACTUALIZACION Y LISTADO DE SECTORES
2630 CLS
2640 WIDTH 40
2650 KEY1,"S"
2660 KEY2,"X"
2670 KEY3,"U"
2680 KEY4,"M"
2690 KEY5,"≈"
2700 CLS:LOCATE 5,1:PRINT"** LISTAR ACTUALIZAR SECTORES **":PRINT:PRINT:PRINT
2710 INPUT" QUE SECTOR QUIERES LISTAR ";SE
2720 PRINT:PRINT:PRINT" QUIERES SALIDA POR IMPRESORA S/N"
2730 A$=INKEY$:IF A$=""THEN 2730
2740 IFA$="S"THEN IM=1ELSE IM=0
2750 IF SE<0 OR SE>720 THEN 2700
2760 PRINT:PRINT:PRINT" F1=PAGINA SIGUIENTE"
2770 PRINT" F2=PAGINA ANTERIOR"
2780 PRINT" F3=ACTUALIZAR PAGINA"
2790 PRINT" F4=MENU"
2800 PRINT" F5=GRABAR SECTOR"
2810 PRINT:PRINT:PRINT" COLOCA EL DISCO Y PULSA UNA TECLA"
2820 IF INKEY$=""THEN2820
2830 CLS:LOCATE10,10:PRINT"PROCESANDO...":AC=0
2840 B$=DSKI$(0,SE)
2850 PO=(PEEK(&HF351)+256*PEEK(&HF352))
2860 IF IM=0 THEN 2990
2870 REM RUTINA DE LISTADO DE SECTORES
2880 LPRINT"SECTOR=";SE;CHR$(11);
2890 FOR N=PO TO PO+496 STEP 16
2900 LPRINT USING "###";N-PO+1;:LPRINT" ";:FORM=0TO15
2910 A$=HEX$(PEEK(N+M)):IF LEN(A$)<2 THEN A$="0"+A$
2920 LPRINTA$;" ";
2930 NEXT
2940 LPRINT" ";
2950 FOR M=0 TO15
2960 A=PEEK(N+M):IFA<&H30 OR A>&H7BTHEN LPRINT".";ELSELPRINTCHR$(A);
2970 NEXT:LPRINTCHR$(11);
2980 NEXT:LPRINTCHR$(12);
```

```
299Ø IF AC=1 THEN RETURN:REM DESPUES DE LA ACTUALIZACION DE UN SECTOR
3ØØØ CLS
3Ø1Ø PRINT" POS SECTOR=";SE
3Ø2Ø FOR LI=Ø TO 19
3Ø3Ø PRINT USING "####";PA+1;:PRINT" ";
3Ø4Ø FOR N=PO+PA TO PO+PA+7
3Ø5Ø A$=HEX$(PEEK(N)):IF LEN(A$)<2 THEN A$="Ø"+A$
3Ø6Ø PRINTA$;" ";
3Ø7Ø NEXT
3Ø8Ø PRINT" ";
3Ø9Ø FOR N=PO+PA TO PO+PA+7
31ØØ A=PEEK(N):IFA<&H3Ø OR A>&H7BTHEN PRINT".";ELSEPRINTCHR$(A);
311Ø NEXT
312Ø PA=PA+8
313Ø IF PA>5Ø5THENLI=19:PA=64Ø
314Ø NEXT
315Ø LOCATEØ,22:PRINT"F1=P.S. F2=P.A. F3=ACT. F4=MENU F5=G.S."
316Ø A$=INKEY$:IF A$=""THEN316Ø
317Ø A=INSTR("SXUM≈",A$):IF A=ØTHEN316Ø
318Ø IF A$="S"THEN IF PA>5Ø5THENPA=48Ø
319Ø IF A$="X"THEN PA=PA-32Ø:IF PA<ØTHENPA=Ø
32ØØ IF A$="M" THEN GOTO 1Ø
321Ø IF A$="U" THEN 324Ø
322Ø IF A$="≈" THEN AC=1:GOSUB286Ø:AC=Ø:DSKO$Ø,SE:RUN
323Ø CLS:GOTO 3Ø1Ø
324Ø X=Ø:Y=1
325Ø SI=PA-16Ø+(X+1+(8*(Y-1)))
326Ø LOCATE2Ø,Ø,Ø:PRINT" POSICION=";SI
327Ø A=STICK(Ø)
328Ø IF A=1THENY=Y-1:IFY<1THENY=1
329Ø IF SI<5Ø5 THENIF A=5THENY=Y+1:IFY>2ØTHENY=2Ø
33ØØ IF SI<512THENIF A=3THENX=X+1:IFX>7THENX=7
331Ø IF A=7THENX=X-1:IFX<ØTHENX=Ø
332Ø LOCATE (X*3+7),Y,1
333Ø A$=INKEY$:IF A$=""THENFORN=1TO1ØØ:NEXT:GOTO325Ø
334Ø IF A$="X" OR A$="S" OR A$="M" OR A$="≈" THEN LOCATE Ø,Ø,Ø:GOTO 318Ø
335Ø A=INSTR("Ø123456789ABCDEF",A$)
336Ø IFA=ØTHEN333Ø
337Ø B$=A$:PRINTA$:LOCATE(X*3+8),Y
338Ø A$=INKEY$:IF A$=""THEN338Ø
339Ø A=INSTR("Ø123456789ABCDEF",A$)
34ØØ IFA=ØTHEN338Ø
341Ø PRINTA$:B$="&H"+B$+A$: LOCATE(32+X),Y:A=VAL(B$):IF A<&H3ØOR
     A>&H7BTHENPRINT"."ELSEPRINTCHR$(A)
342Ø POKE(PO+PA-16Ø+(X+(8*(Y-1)))),A
343Ø IF SI<512THENX=X+1:IFX>77THENX=Ø:Y=Y+1:IFY>2ØTHENY=1
344Ø GOTO 325Ø
345Ø DEFUSR=Ø:A=USR(A)
346Ø REM LISTAR UN PROGRAMA
347Ø DI=PEEK(&HF351)+256*PEEK(&HF352)
```

```
348Ø CLS:X=5:FILES:PRINT
349Ø PRINT
35ØØ INPUT"PROGRAMA A BUSCAR=";P$
351Ø XX=4
352Ø XX=XX+1:IF XX>11 THEN PRINT"FIN DE DIRECTORIO:FORN=1TO3ØØØ:NEXT:RUN
353Ø A$=DSKI$(1,XX)
354Ø FOR I=DI TO DI+48Ø STEP 32
355Ø A$=""
356Ø FOR J=Ø TO 1Ø
357Ø IF J=8 AND CHR$(PEEK(I+J))<>" " THEN A$=A$+"."
358Ø IF CHR$(PEEK(I+J))<>"" THENA$=A$+CHR$(PEEK(I+J))
359Ø NEXT J
36ØØ CL=PEEK(I+26)+256*PEEK(I+27)
361Ø IF CL=Ø THEN PRINT"**** NO HE ENCONTRADO "P$" ****":FORN=1TO 3ØØØ:NEXT:RUN
362Ø PRINTA$:IF A$<>P$THEN387Ø
363Ø SE=(CL-2)*2+12:LO=(PEEK(I+28)+256*PEEK
     (I+29))+256*(PEEK(I+3Ø)+256*PEEK(I+31))
364Ø CLS
365Ø PRINT"***LOCALIZADO***":PRINTA$:PRINT
366Ø CL$="ØØØØ"+HEX$(CL):CL$=RIGHT$(CL$,4)
367Ø SE$="ØØØØ"+HEX$(SE):SE$=RIGHT$(SE$,4)
368Ø LO$="ØØØØ"+HEX$(LO):LO$=RIGHT$(LO$,4)
369Ø PRINT" CLUSTER=";:PRINTUSING"#########";CL:PRINT;"-";CL$;"H"
37ØØ PRINT" SECTOR =";:PRINTUSING "#########";SE;:PRINT"-";SE$;"H":PRINT"
     LONGI.=";:PRINTUSING"########,";LO;:PRINT"-";LO$;"H"
371Ø PRINT
372Ø A$=DSKI$(1,SE)
373Ø IF PEEK(DI)=255THENPRINT"BASIC"ELSEIFPEEK(DI)=254THENPRINT"COD.MAQ."
     ELSEIFPEEK(DI)THENPRINT"A S C I I "ELSEPRINT"*DESCO*"
374Ø IF PEEK(DI)<>254THENGOTO384Ø
375Ø D1=(PEEK(DI+1)+256*PEEK(DI+2))
376Ø D2=(PEEK(DI+3)+256*PEEK(DI+4))
377Ø D3=(PEEK(DI+5)+256*PEEK(DI+6))
378Ø D1$="ØØØØ"+HEX$(D1):D1$=RIGHT$(D1$,4)+"H"
379Ø D2$="ØØØØ"+HEX$(D2):D2$=RIGHT$(D2$,4)+"H"
38ØØ D3$="ØØØØ"+HEX$(D3):D3$=RIGHT$(D3$,4)+"H"
381Ø PRINT:PRINT" DIR.COM= ";:PRINTUSING"#####,";D1;:PRINT"-";D1$
382Ø PRINT" DIR.FIN= ";:PRINTUSING"#####,";D2;:PRINT"-";D2$
383Ø PRINT" DIR.EJE= ";:PRINTUSING"#####,";D3;:PRINT"-";D3$
384Ø PRINT:PRINT:PRINT"QUIERES SEGUIR BUSCANDO S/N?"
385Ø A$=INKEY$:IF A$="" THEN 385Ø ELSE IF A$<>"S"ANDA$<>"s"ANDA$<>" " AND
     A$<> CHR$(13) THEN GOTO 1Ø
386Ø GOTO346Ø
387Ø NEXT
388Ø GOTO352Ø
```

El programa está dividido en partes totalmente autónomas y bien diferenciadas. De la línea 10 a la 290 tenemos la presentación y el menú de opciones, a continuación y hasta la línea 1550, la rutina de listados de información de los ficheros en el disco.

De la línea 470 a la 800 se lee el directorio, se calcula el cluster y sector de comienzo y se guarda todo esto en la matriz DI$, seguidamente se clasifica esta matriz por número de sector lógico, a fin de optimizar

posteriormente los movimientos de las cabezas de lectura, al buscar en los ficheros su tipo y direcciones (líneas 850 a 1010); para terminar se clasifica nuevamente por nombre de fichero y se listan los resultados en las líneas 1080 a 1550. Seguidamente, tenemos una clásica rutina de imprimir pantallas (HARDCOPY) de la línea 1560 a 1570, estando esta opción siempre disponible mediante la tecla F10.

La rutina que nos muestra el mapa de ocupación del disco, está en las líneas 1630 a 1970; esta información se obtiene procesando la F.A.T. La rutina que comienza en la línea 2040, es la encargada de preparar e imprimir la F.A.T. de una manera inteligible.

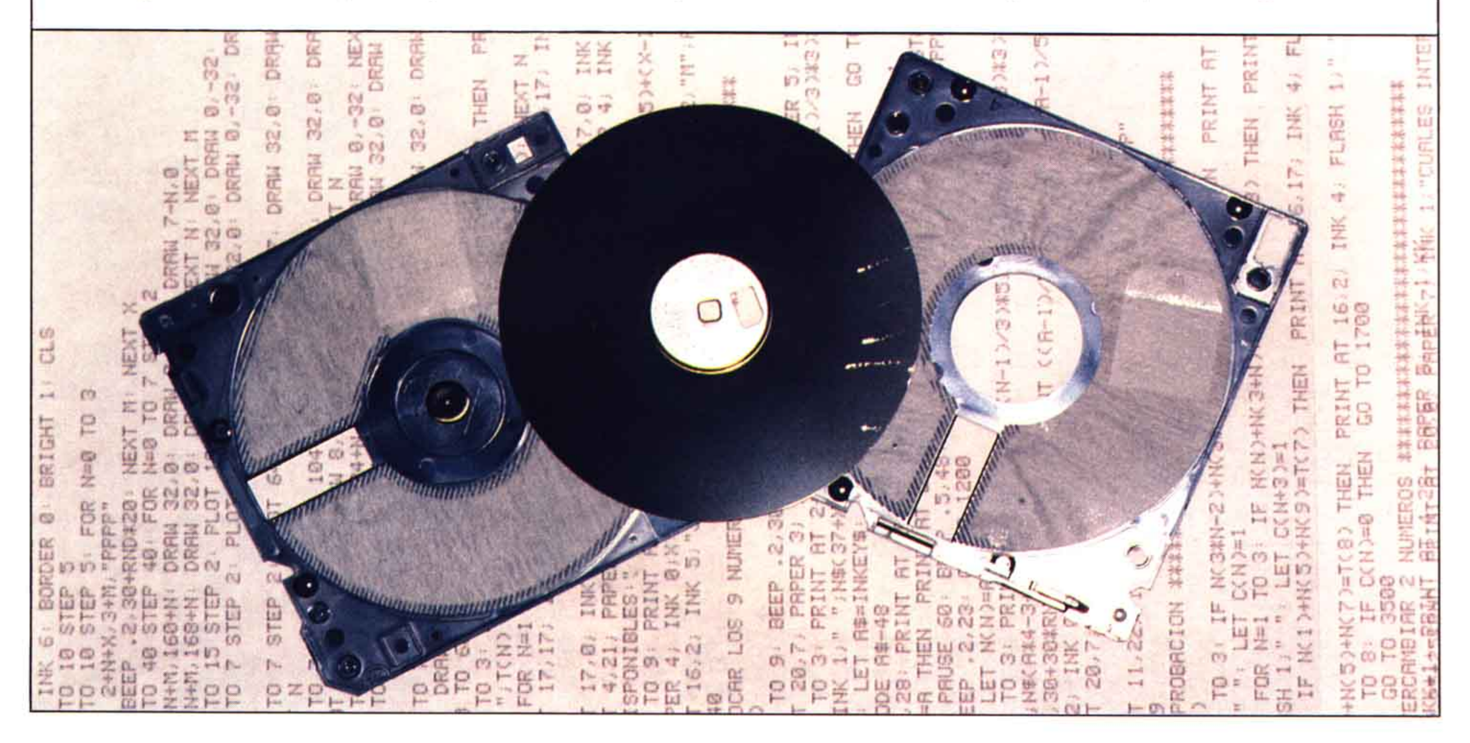
A continuación tenemos la rutina de listar y actualizar sectores, desde la línea 2520 a la 3340. Esta rutina comienza asignando las teclas de función para permitirnos mediante éstas, el avanzar o retroceder páginas, actualizar una pantalla, o grabar un sector actualizado.

F1 = PAGINA ANTERIOR
F2 = PAGINA SIGUIENTE
F3 = ACTUALIZAR PAGINA
F4 = MENU PRINCIPAL
F5 = GRABAR SECTOR

A continuación, nos pide el número del sector que deseamos visualizar y lo muestra en pantalla. Ya que un sector tiene 512 posiciones y no caben todas en pantalla, nos lo muestra en varias pantallas, listándolo en hexadecimal y en ASCII (2900 a 3050). En las líneas 3060 a 3120, el programa espera a que se pulse una tecla de función, mostrándonos la siguiente pantalla (F2), o pasando a actualizar la presente si pulsamos F3. En ese momento, aparece el cursor en pantalla y nos muestra su posición dentro del sector, pudiéndolo desplazar con las teclas de cursor. Cuando alcancemos la posición deseada, escribiremos su valor hexadecimal, mostrándose inmediatamente en el lado derecho de la pantalla su traducción ASCII (líneas 3100 a 3340); al terminar con todas las actualizaciones pulsaremos F5 para grabar el sector en el disco.

Por último, en las líneas 3370 a 3790 tenemos la rutina de listar información de un fichero. Esta rutina nos muestra los programas que hay en el disco y nos pide el nombre de uno de ellos, cuando lo localiza nos muestra el nombre, tipo, longitud y las direcciones si es código máquina. Se puede utilizar esta entrada del menú, cuando queremos información solamente de un fichero determinado, siendo bastante más rápida que la primera de las opciones.

El programa es bastante interactivo haciendo preguntas, por lo que no tendréis problemas de manejo. Solamente recordar, que si actualizáis la F.A.T. o el directorio, podéis perder toda la información que tengáis en el disco, por lo que es conveniente que antes de tocar nada, hagáis una copia de seguridad.



SISTEMAS DE N ECUACIONES CON N INCOGNITAS

En un anterior artículo os mostramos un programa que resolvía «sistemas de 3 ecuaciones con 3 incógnitas», recordamos los métodos tradicionales y seguimos los pasos del programa; pero se nos planteó una duda: ¿cómo resolver un sistema de N ecuaciones con N incógnitas?

El programa que adjuntamos en aquella ocasión no era capaz de resolver este problema, porque para hacerlo se hubiera tenido que aplicar el método de la «suma de los menores adjuntos» para hallar los determinantes, ya que la dimensión de las matrices sería superior a 3.

Por ello hemos elaborado el que aquí os presentamos, que incluye una serie de instrucciones que efectúan estos cálculos, simplificándolos hasta llegar a una matriz de 2 × 2 o 3 × 3, donde podremos hallar fácilmente los determinantes.

Este proceso nos permitirá aplicar la teoría de Cramer, con la que solucionaremos el problema.

Para simplificar el cálculo en el programa, nos hemos basado en rutinas recursivas.

RUTINAS RECURSIVAS

Las Rutinas Recursivas son las que se llaman a ellas mismas, siendo reiniciadas antes de acabar la ejecución.

Se dice que una rutina es «reentrante» cuando puede ser llamada por ella misma antes de finalizar. Por tanto, las rutinas pueden ser «reentrantes» y no recursivas a la vez, ya que en éstas existe una mayor restricción.

Toda rutina recursiva debe cumplir dos requisitos:

– que disponga de una condición de FIN

– que esta condición llegue a cumplirse después de un cierto número de iteraciones.

Seguidamente veamos un ejemplo en el que no se han tenido en cuenta dichos condicionantes:

```
5 REM PROGRAMA PRINCIPAL
1Ø A=Ø
2Ø GOSUB 1ØØ
3Ø END
1ØØ REM RUTINA QUE SE
    LLAMA A ELLA MISMA
11Ø A=A+1
12Ø GOSUB 1ØØ
13Ø RETURN
```

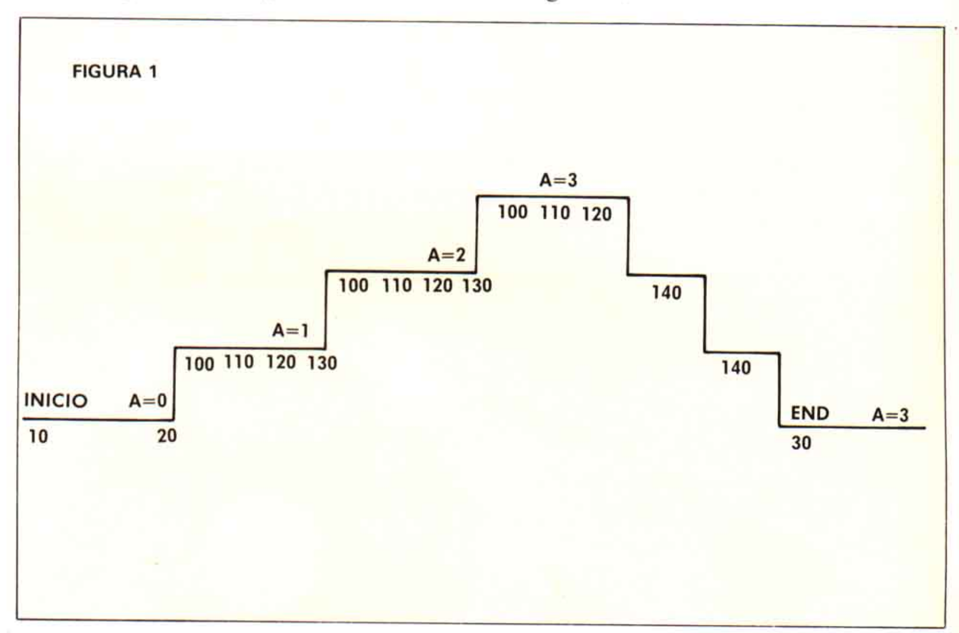
Como podréis comprobar, al no disponer de una condición de salida, la rutina entra en un bucle sin fin hasta producir un error en el sistema.

Dicho error es provocado al sobrepasar la capacidad de memoria de la máquina, ya que en cada llamada a la subrutina se debe guardar la dirección de retorno en el stack, función que puede realizar un número limitado de veces.

Por el contrario, veamos qué ocurre si añadimos una condición que limite el número de iteraciones:

```
5 REM PROGRAMA PRINCIPAL
1Ø A=Ø
2Ø GOSUB 1ØØ
3Ø END
1ØØ REM RUTINA QUE SE
    LLAMA A ELLA MISMA
11Ø A=A+1
12Ø IF A≥3 THEN RETURN
13Ø GOSUB 1ØØ
14Ø RETURN
```

La secuencia de instrucciones sería la siguiente:



No sólo es este planteamiento el correcto, sino que podemos llegar a la misma conclusión empleando otras secuencias de instrucciones, como observaréis en el ejemplo siguiente:

```
5 REM PROGRAMA PRINCIPAL
1Ø A=Ø
2Ø GOSUB 1ØØ
3Ø END
```

- SURGE UNA DUDA
- APLICACION DE LA TEORIA DE CRAMER
- RUTINAS RECURSIVAS
- FACTORIALES

- ESTRUCTURAS DO-WHILE Y FOR-NEXT
- DESCRIPCION DEL PROGRAMA
- VARIABLES LOCALES Y GLOBALES

```
1ØØ  RUTINA QUE SE LLAMA A
     ELLA MISMA
11Ø  A=A+1
12Ø  IF A<3 THEN GOSUB 1ØØ
13Ø  RETURN
```

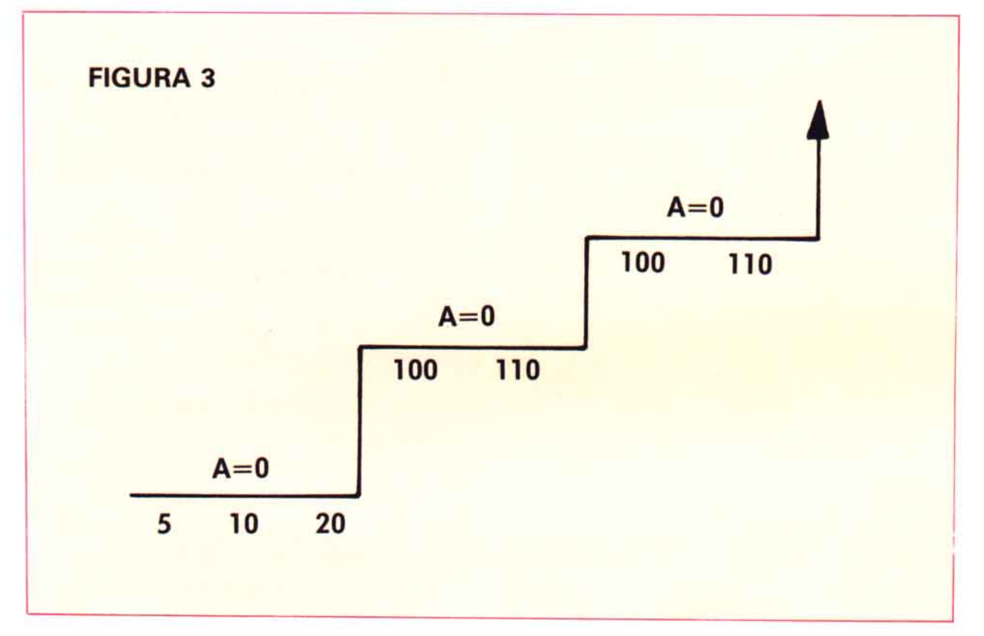
FIGURA 2

Al igual que en el anterior caso, también funciona. El proceso es como sigue: se van anidando subrutinas hasta encontrar la condición de FIN de la línea 120. Seguidamente deshace todas las llamadas por la ejecución encadenada de las instrucciones RETURN, que hallamos en la línea 140 en el primer ejemplo; y en la 130, en el segundo.

Otro posible error que podemos cometer al construir el programa con Rutinas Recursivas es incluir la instrucción de salida pero que ésta no llegue a ejecutarse nunca.

```
5  REM PROGRAMA PRINCIPAL
1Ø  A=Ø
2Ø  GOSUB 1ØØ
3Ø  END
1ØØ  REM RUTINA QUE SE
     LLAMA A ELLA MISMA
11Ø  IF A<3 GOSUB 1ØØ
12Ø  A=A+1
13Ø  RETURN
```

En la línea 10, la variable A se hace igual a 0. En la línea 110 al cumplir con la condición de que A es menor que 3, vuelve a saltar sobre ella misma sin llegar a pasar por la línea 120, donde iría incrementando la variable A.



En la secuencia de instrucciones es donde se muestra más claramente el problema, que se repetiría hasta causar error en el sistema.

FACTORIALES

El factorial de un número es la cantidad que resulta de multiplicar el número por sus anteriores sucesivos hasta llegar a la unidad.

Se expresa del siguiente modo:

$$N! = \prod_{j=1}^{j=N} j$$

Un ejemplo numérico sería:

5!= 5 * 4 * 3 * 2 * 1 = 120

Proseguimos con un programa con el que podemos calcular este número, utilizando un bucle FOR-NEXT.

```
10 N=5
15 R=1
20 FOR J=N TO 1 STEP -1
30 R=R*J
40 NEXT J
50 PRINT "EL FACTORIAL DE ";
   N;"ES: ";R
60 END
```

Otra manera de realizar este cálculo, sería usando rutinas recursivas:



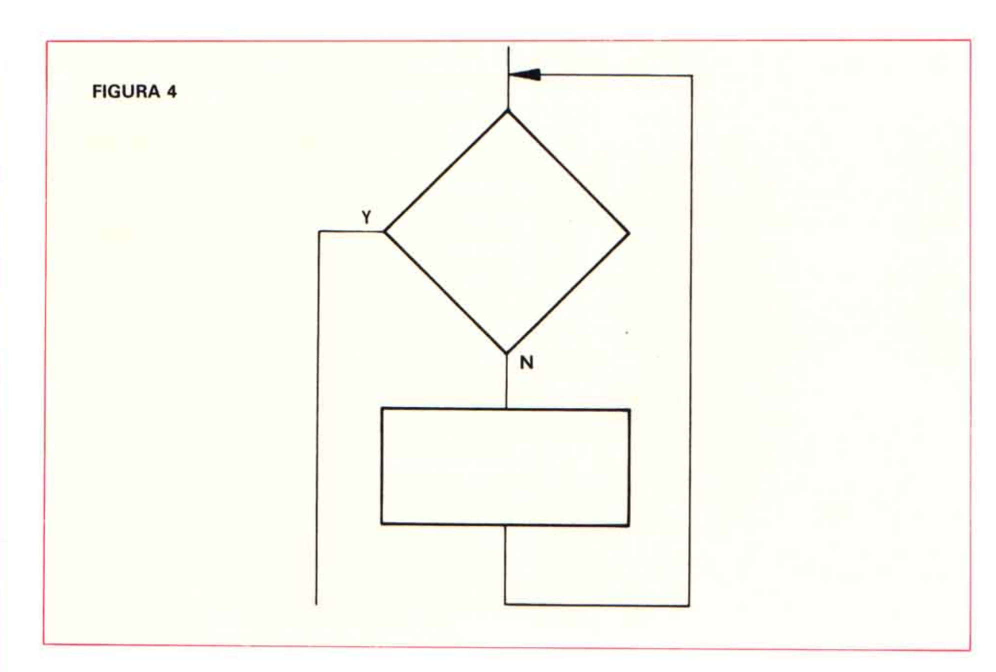
```
10 N=5
15 J=N:R=1
20 GOSUB 100
30 PRINT "EL FACTORIAL DE ";
   N;"ES: ";R
40 END
100 REM CALCULO
    FACTORIAL
110 R=R*J
120 J=J-1
130 IF J>1 THEN GOSUB
    100
140 RETURN
```

ESTRUCTURAS DO-WHILE Y FOR-NEXT

En todo cálculo iterativo se puede realizar un programa que represente al algoritmo mediante dos estructuras básicas, que en apariencia son bastante parecidas.

La primera corresponde a la estructura DO-WHILE, que se representa con tal diagrama:





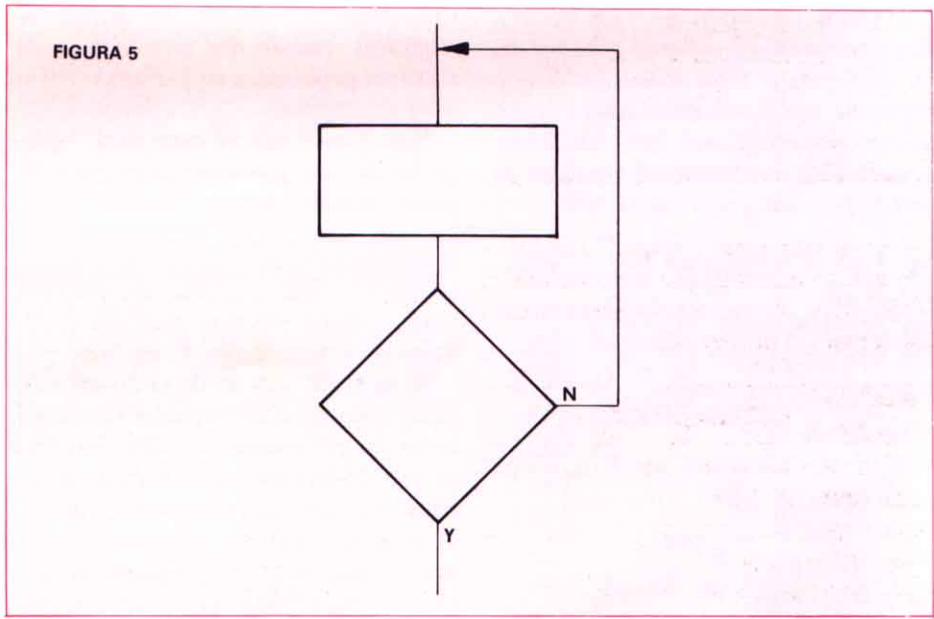
cuando realmente se producen los cálculos del resultado.

Como observaréis, cuando la variable N es igual a 1, no ejecuta la instrucción 140, pasando directamente a escribir el valor en la pantalla.

```
10 REM PROGRAMA PRINCIPAL
15 N=3
20 GOSUB 100
30 PRINT F(N)
40 END
100 IF N=1 THEN F(N)=1:GOTO
    150
110 N=N-1
120 GOSUB 100
130 N=N+1
140 F(N)=N*F(N-1)
150 RETURN
```

La segunda corresponde al ya conocido bucle FOR-NEXT, representado por el siguiente ordinograma:





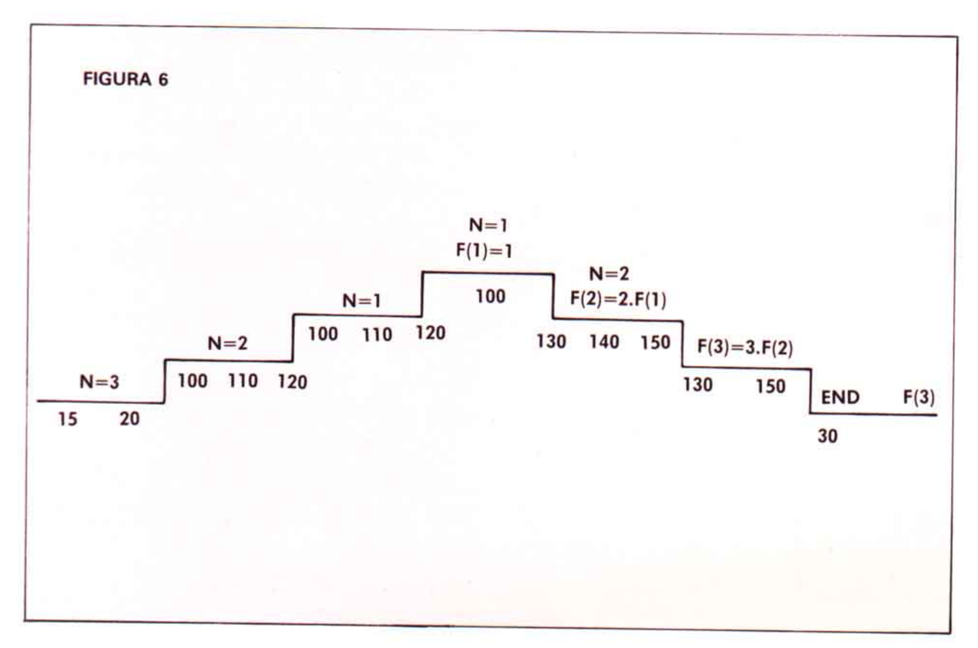
Pese a su similitud, existe una gran diferencia básica entre ellas, y es que el bucle FOR-NEXT siempre se ejecuta como mínimo una vez, mientras que la estructura DO-WHILE puede no hacerlo en ninguna ocasión si de entrada no se cumple esta condición.

Esto mismo sucedería si en una rutina recursiva hacemos la pregunta para hallar la condición de fin, antes del proceso o después del mismo.

Así, en el ejemplo que figura a continuación, el funcionamiento sería en primer lugar llamar a la rutina, solamente buscando la última llamada, y durante los sucesivos retornos es

Una rutina alternativa a la anterior es:

```
10 REM PROGRAMA PRINCIPAL
20 N=3
30 GOSUB 100
40 PRINT F(N)
50 END
100 N=N-1
110 IF N=1 THEN F(N)=1:GOTO
    150
120 GOSUB 100
140 F(N)=N*F(N-1)
150 N=N+1
160 RETURN
```



Habréis advertido que los resultados intermedios los hemos conservado en una matriz. Esto lo hemos hecho para una mayor claridad en el seguimiento del programa, pero no es necesario, así que vamos a sustituir la matriz por una única variable R.

```
10 REM PROGRAMA PRINCIPAL
20 N=3
30 GOSUB 100
40 PRINT F(N)
50 END
100 N=N-1
110 IF N=1 THEN F(N)=1:GOTO 150
120 GOSUB 100
140 F(N)=N*F(N-1)
150 N=N+1
160 RETURN
```

Todos los cálculos los hemos efectuado sobre el sencillo ejemplo de hallar el factorial de un número; aunque en este caso no es necesario el empleo de rutinas recursivas, ya que incluso hubiera resultado más corto y claro si hubiéramos empleado un bucle FOR-NEXT que nos hubiera resuelto el problema fácilmente. Pero hay que tener en cuenta que en nuestro cometido precisábamos de un ejemplo sencillo al que poder aplicar las teorías de la recursividad.

En la realidad, existen otros tipos de subrutinas en las cuales sí es muy interesante el empleo de dichas estructuras, puesto que son capaces de resolver problemas en planteamientos muy concretos.

Éste puede ser el caso de la resolución de *determinantes de matrices de grado superior a 3.*

DESCRIPCION DEL PROGRAMA

En la línea 330, nos pregunta el número de ecuaciones e incógnitas.

Si os fijáis, no se delimita en ninguna condición un número determinado de ecuaciones e incógnitas, así de esta forma no limita el número de ellas; aunque evidentemente y aunque el programa parezca corto, el tiempo que transcurrirá calculando variará según la complejidad de los cálculos que deba realizar.

Una vez introducidos los datos, así como el término independiente, los analiza, calculando el número total de determinantes de 2 * 2 que deberá efectuar, rutina que comienza en la línea 560 a 640. Dichos cálculos se llevan a cabo para podernos informar de qué tanto por ciento del trabajo de cálculo se lleva efectuado durante el largo proceso de hallar la solución del sistema de ecuaciones.

Seguidamente, en la línea 650 escribirá en la pantalla «Proceso de Cálculo», pasando de inmediato a calcular el determinante general.

Si el valor de dicho determinante es igual a 0, en la línea 730 escribirá «Sistema sin solución» para continuar en la 750 imprimiendo «Las ecuaciones son linealmente dependientes», finalizando aquí el programa.

En caso contrario, hallará el determinante particular de cada variable, rutina que comienza en la línea 780 a 850.

Dentro de esta subrutina se efectuarán dos saltos, uno en la línea 810 a la línea 1020 donde se halla la rutina Swap; y otro en la línea 820 a la 1080, donde está la rutina de calcular el determinante.

La rutina Swap es la encargada de efectuar las sustituciones, es decir, intercambia las columnas de la matriz principal, movimientos que son necesarios para poder hallar el determinante particular de cada coeficiente.

La rutina de cálculo, como imaginaréis, es de las más importantes del programa. Ocupa desde las líneas 1080 a la 1360.

¿Recordáis que casi al comienzo del programa hemos dicho que había una rutina que calculaba el número de determinantes de 2 * 2 que iba a realizar? Pues en la línea 1370 a 1450 se halla una rutina encargada de escribir en la pantalla el porcentaje de operaciones efectuadas cada vez que se calcula un determinante.

Una vez obtenidos los determinantes, realizará el cálculo final para hallar los resultados. Esto ocurre en las líneas 860 a 900; para pasar, por último, a la impresión de los resultados del sistema en la rutina de las líneas 910 a 1010 que pone el punto y final al programa.

Para los que queráis seguir el programa os diremos que las variables están guardadas en sucesivas matrices de dos dimensiones, tomando «pisos» apilados.

En cada piso se guardan las variables para el cálculo de una de las dimensiones de la matriz. Dicha matriz cúbica es de la misma dimensión que el sistema a resolver.

Nos hemos visto obligados a emplear este sistema, por no disponer de variables locales en el lenguaje BASIC, que en esta aplicación muestra

sus limitaciones; disponiendo tan solamente de variables globales.

Una variable es global cuando se puede leer o modificar desde cualquier punto del programa; mientras que una variable local sólo existe dentro de la subrutina donde se ha definido, desapareciendo al final de ésta.

```
250 KEY OFF
260 CLS
270 PRINT" RESOLUCION DE
    SISTEMAS LINEALES"
280 PRINT
290 PRINT" DE N ECUACIONES
    Y N INCOGNITAS."
300 PRINT"-----------------------------
    ---"
310 PRINT:PRINT
320 CLEAR
330 INPUT "Numero de
    ecuaciones : ":D
340 PRINT:PRINT
350 DIM M(D.D,D)
360 DIM S(D)
370 DIM I(D)
380 DIM P(D)
390 DIM J(D)
400 DIM K(D)
410 DIM SC(D)
420 DIM DT(D)
430 DIM T(D)
440 DIM R(D)
450 'entrada de datos
460 '---------------
    ---
470 FOR I=1 TO D
480 PRINT "Ecuacion";I
490 PRINT "--------------"
500 FOR J=1 TO D
510 PRINT "valor de
    ":CHR$(J+96):" : ";:INPUT
    M(D,I,J)
520 NEXT J
530 INPUT "Termino ind :";T (I)
540 PRINT
550 NEXT I
560 'num de det de 2*2
570 '---------------
    ----
580 N=3
590 IF D=2 GOTO 630
600 FOR J=4 TO D+1
610 N=N*J
620 NEXT J
630 ND=N:NP=0
640 CLS
650 PRINT" PROCESO DE
    CALCULO.
660 PRINT" -----------
    ----------
670 'calculo del determinante
    gral
680 '-----------------------
    ------------
690 GOSUB 1080
700 DT=SC(D)
710 IF DT<>0 GOTO 780
720 CLS
730 PRINT "Sistema sin solucion."
740 PRINT
750 PRINT "Las ecuaciones son
    linealmente dependientes."
760 PRINT
770 END
780 'calc de determinantes de
    coef
790 '-----------------------
    ------------
800 FOR I=1 TO D
810 GOSUB 1020
820 GOSUB 1080
830 DT(I)=SC(D)
840 GOSUB 1020
850 NEXT I
860 'calculo de resultados
870 '-----------------------
    --
880 FOR I=1 TO D
890 R(I)=DT(I)/DT
900 NEXT I
910 'impresion de resultados
920 ' -----------------------
    -----
930 CLS
940 PRINT" RESULTADOS DEL
    SISTEMA.
950 PRINT" ----------------
    -----------
960 PRINT:PRINT
970 FOR I=1 TO D
980 PRINT"variable";I; "=";R(I)
990 NEXT I
1000 PRINT:PRINT
1010 END
1020 'subrutina swap
1030 ' ---------------
1040 FOR J=1 TO D
1050 SWAP M(D,J,I),T(J)
1060 NEXT J
1070 RETURN
1080 'calculo del determinante
1090 ---------------------
     -------
110 IF D>2GOTO 1150
1110 NP=NP+1
1120 GOSUB 1370
1130 SC(2)=((M(2,1,1) * M (2,
     2,2)) - (M(2,1,2)*M(2,2,1)))
1140 GOTO 1360
1150 M(D,0,0)=0
1160 S (D) = 1
1170 FOR R=1 TO D
1180 P(D)=1
1190 FOR S=1 TO D-1
1200 IF S=R THEN
     P(D)=P(D)+1)
1210 FOR T=1 TO D-1
1220 M(D-1,S,T) = M(D,P(D),
     T+1)
1230 NEXT T
1240 P(D)=P(D)+1
1250 NEXT S
1260 I(D)=R: J(D)=S: K(D)=T
1270 D=D-1
1280 GOSUB 1080
1290 D=D+1
1300 R=I(D): S=J (D):T=K(D)
1310 M(D,R,O) = M(D,R,1)*SC
     (D-1)*S(D)
1320 S(D)=S(D) *(-1)
1330 M(D,O,O)=M(D,O,O) +
     M(D,R,O)
1340 NEXT R
1350 SC(D)=M(D,O,O)
1360 RETURN
1370 'imprime % realizado
1380 ' ---------------------
     -
1390 TP=100*(NP/ND)
1400 TP=INT(100*TP)/100
1410 LOCATE 0,2
1420 PRINT" "
1430 LOCATE 0,2
1440 PRINT"% =";TP
1450 RETURN
```

LAS VARIABLES DEL SISTEMA (I)

Hemos visto en anteriores artículos de INPUT una breve panorámica del microprocesador Z80 y de la arquitectura interna del ordenador MSX.

En el capítulo de este mes nos centraremos en el estudio de las principales variables internas del sistema y en sucesivos números estudiaremos en profundidad la ROM (Read Only Memory) y aprenderemos a utilizar algunas eficaces subrutinas.

La tabla de las variables internas del sistema se denomina también *Zona de trabajo de la BIOS* (Basic Input Output System). Dicha zona de trabajo comienza en la dirección hexadecimal F380 (62336 en decimal) y existe en todas las configuraciones de los MSX, independientemente de su capacidad de memoria (16, 32, 64 k).

DE HEXADECIMAL A DECIMAL

Por razones de comodidad y aprendizaje presentaremos las direcciones de las variables del sistema en base numérica hexadecimal (base 16). Si todavía no estamos acostumbrados a utilizarla y deseamos una traducción a base numérica decimal, solamente hará falta escribir en BASIC: PRINT &HXXXX (donde XXXX es cualquier dirección en hexadecimal) y RETURN. El BASIC nos devolverá un valor negativo; para obtener su dirección real positiva solamente nos hará falta sumar 65536.

Así, en el ejemplo que a continuación te presentamos, la dirección hexadecimal &HF6AA (*flag* de AUTO) se descompondrá de la siguiente forma:

PRINT &HF6AA <RETURN>
–2390
PRINT –2390 + 65536
<RETURN> 63146

Otra forma para hallar la dirección en decimal de un número en hexadecimal sería hacer lo siguiente:

N.º DECIMAL = PRINT (10 * 1 + 10 * 16 + 6 * 256 + 4096 * 15)
<RETURN> 63146

Las variables del sistema se pueden dividir en dos grupos principales:

a) Las variables que contienen un *flag* (indicador) o una constante comprendida entre 0 y 255. Dichas variables poseen un solo octeto (byte).

b) Las variables que contienen una dirección (16 bits) o un valor comprendido entre 0 y 65535. Dichas variables están constituidas por dos octetos consecutivos.

LECTURA Y ESCRITURA DE LAS CONSTANTES

a) *Constantes de 1 octeto (8 bits):*

Las constantes de un solo byte y los *flags* se leen por medio de una simple función PEEK y se escriben con la ayuda de una instrucción POKE.

Ejemplos:

Leer el contenido de la dirección de memoria F3E9 (color del texto):

PRINT PEEK (&HF3E9)
<RETURN> 15

Modificar el contenido de la variable que contiene el número de color de texto, inscribiendo el número 6:

POKE &HF3E9,6

En lenguaje ensamblador el ejemplo anterior se puede formular de dos modos:

— lectura directa
```
LD A,(F3E9H)
```
— lectura indirecta
```
LD HL,(F3E9H)
LD A,(HL)
```
— escritura directa
```
LD A,6
LD (F3E9H),A
```
— escritura indirecta
```
LD A,6
LD HL,F3E9H
LD (HL),A
```



b) *Constantes de 16 bits:*

Las constantes de 16 bits son un poco más difíciles de utilizar y manipular.

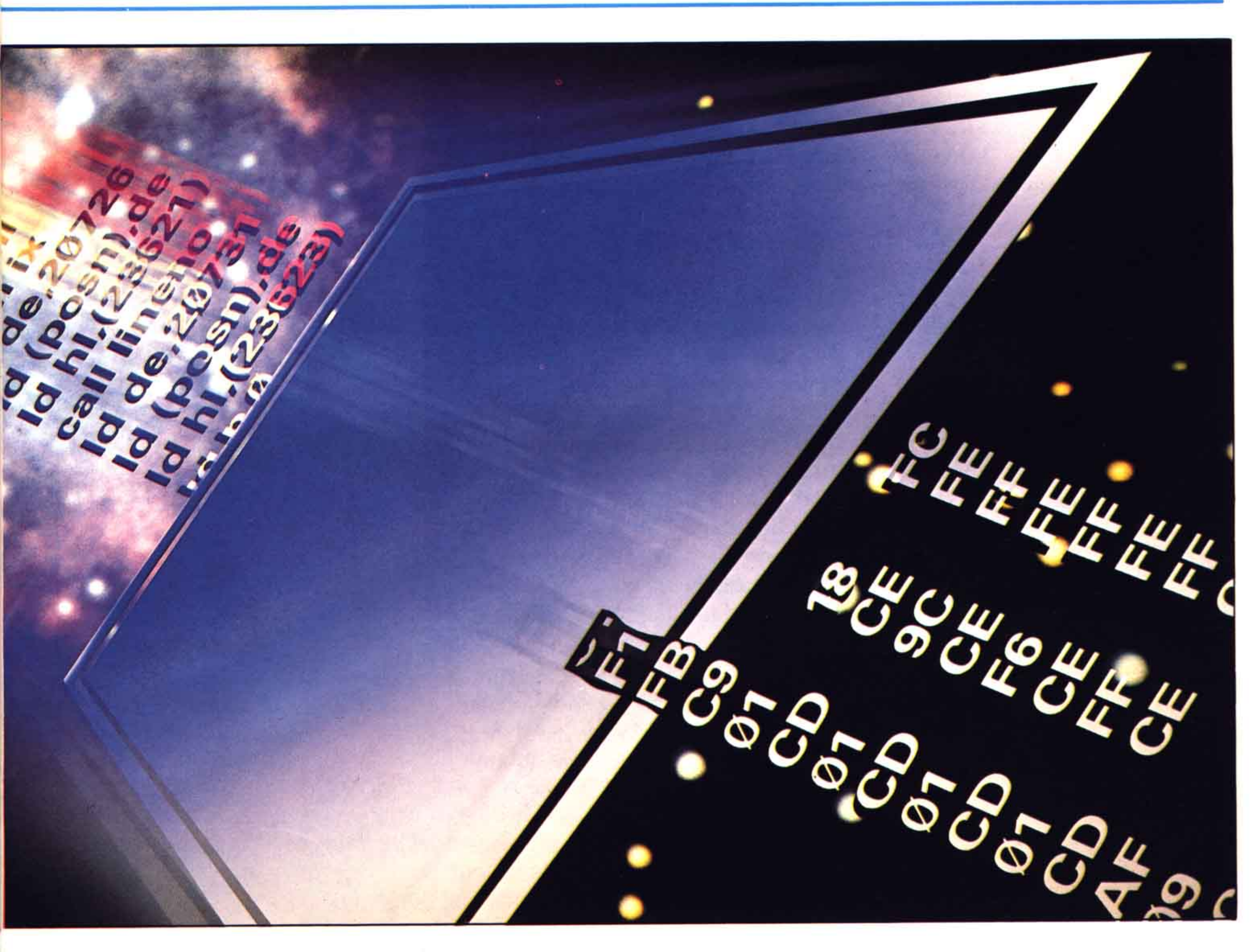
Ejemplo:

Leer la dirección contenida en la dirección FCBF (dirección de ejecución de un programa cargado por BLOAD):

PRINT PEEK (&HFCBF)+256 * PEEK (&HFCC0) el resultado final será en decimal; si lo queremos en hexadecimal haremos lo siguiente:

PRINT HEX$ (PEEK(&HFCBF)

- LA ZONA DE TRABAJO DE LA BIOS
- DE HEXADECIMAL A DECIMAL
- LECTURA Y ESCRITURA DE LAS CONSTANTES
- LAS VARIABLES INTERNAS DEL SISTEMA MSX
- CONSTANTES DE 8 Y DE 16 BITS
- RELACION DE LAS PRINCIPALES VARIABLES



+ 256 * PEEK(&HFCC0))

En lenguaje ensamblador la escritura/lectura de estas constantes o direcciones no pasa jamás por el acumulador (tal como hacían las de 8 bits) sino por un par de registros: HL, DE, BC.

Así el ejemplo anterior posee dos métodos de lectura:

— lectura directa

```
LD HL, (FCBFH) ;el resultado
se encuentra en HL
```

— lectura indirecta

```
LD HL,FCBFH
LD E,(HL)
INC HL
LD D,(HL) ;el resultado se en-
cuentra en DE
```

La escritura de constantes de 16 bits en lenguaje ensamblador posee también los métodos directo e indirecto:

— método de escritura directa:

```
LD HL, C800H
LD FCBFH,HL
```

— método de escritura indirecta:

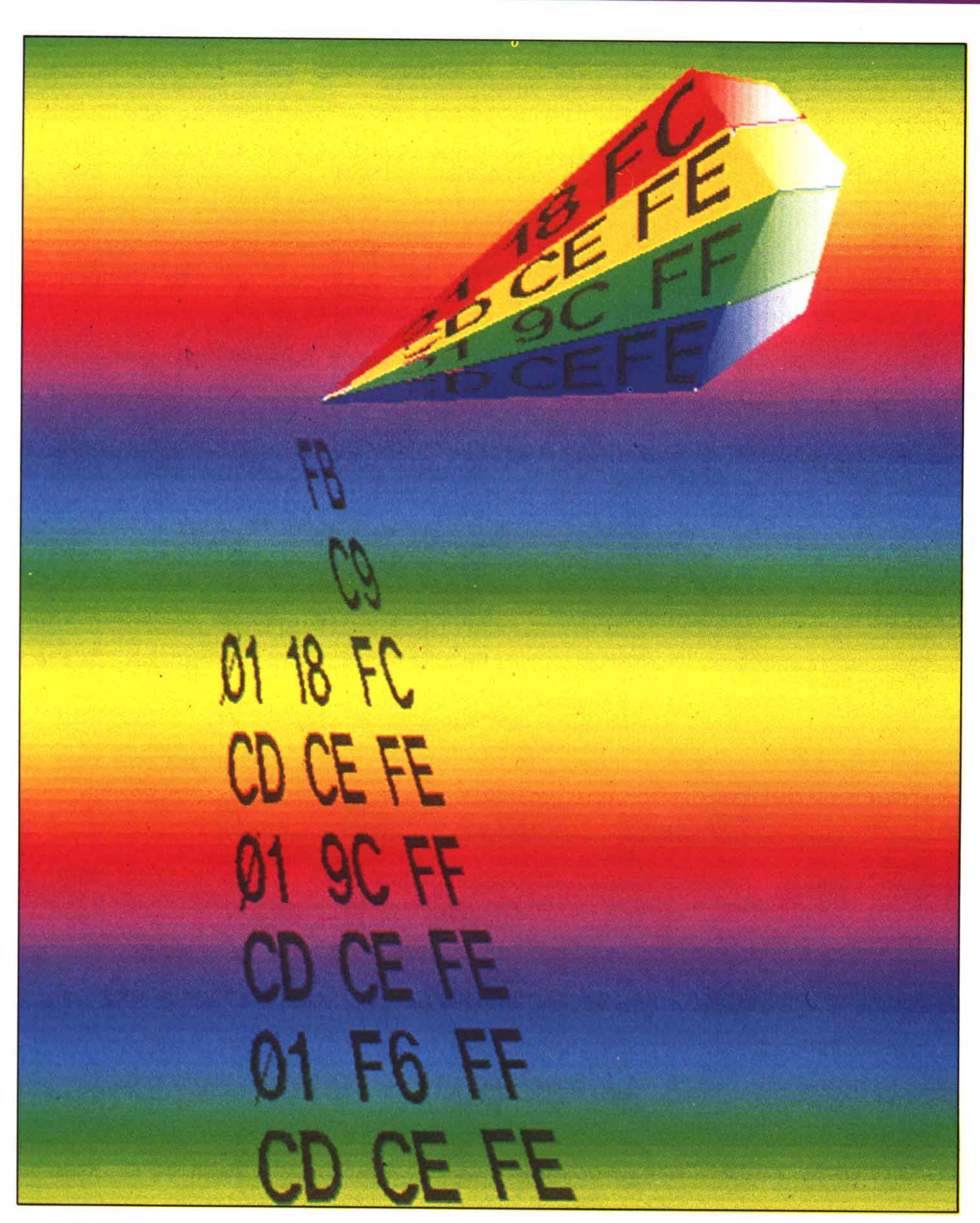
```
LD HL,FCBFH
LD DE,C800H
LD (HL),E
INC HL
LD (HL),D
```

VARIABLES DEL SISTEMA

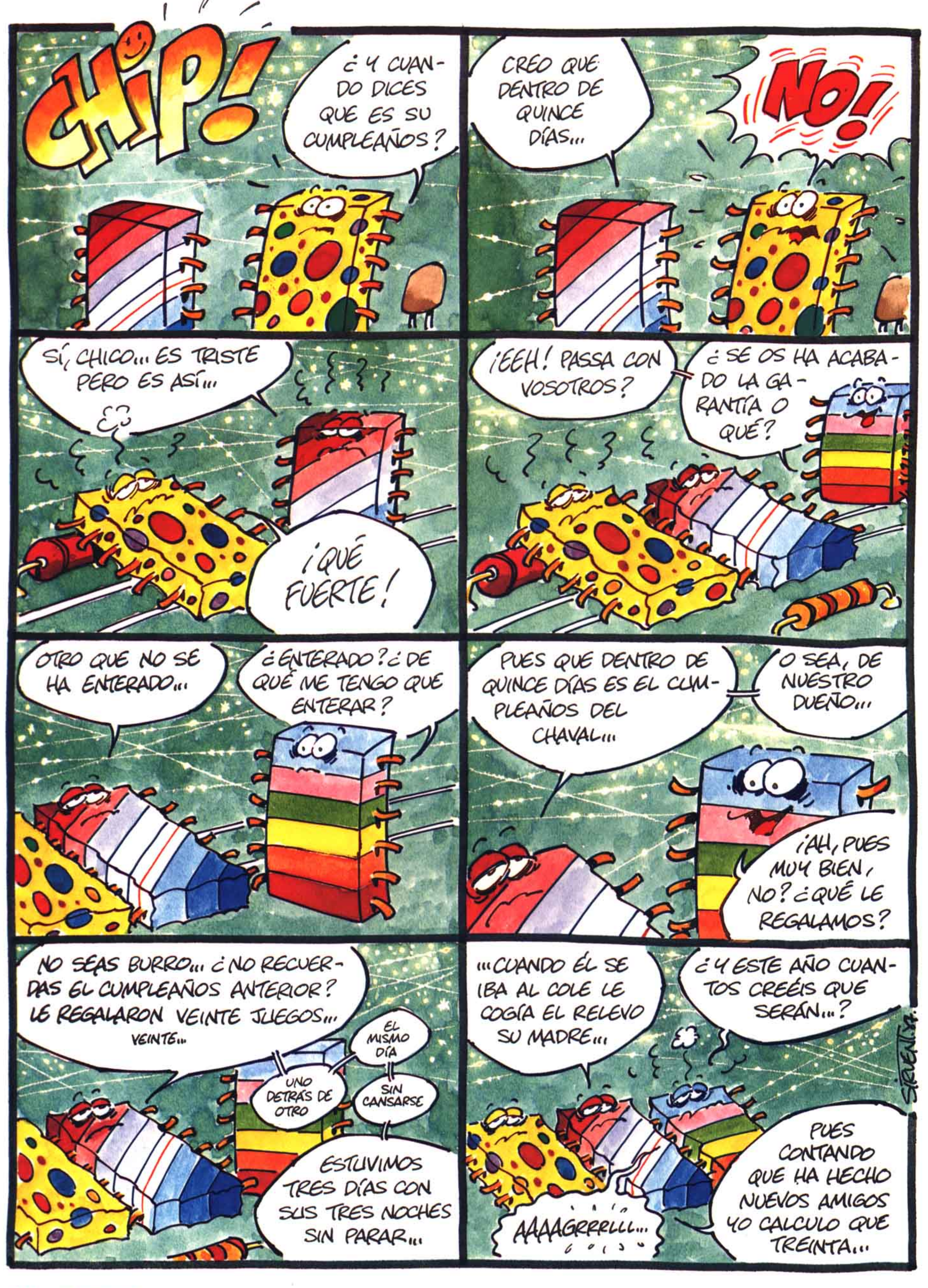
A continuación os presentamos algunas de las principales variables internas del sistema MSX. Con ellas podréis obtener espectaculares efectos tanto en BASIC como en lenguaje ensamblador. Busca, compara y prueba y si no veis lo que os interesa aguardad al próximo capítulo de *Código máquina al alcance de todos*.

Direc. hexadecimal inicio	fin	Nombre	Función
F38Ø	F384	SWROM	Rutina de lectura de los slots del banco Ø
F385	F38B		Rutina de escritura de los slots del banco Ø
F38C	F399	JPROM	Rutina de salto al interior de un slot del banco Ø
F39A	F3AD	USRTAB	Tabla de las direcciones definidas por la instrucción DEFUSRn=
F3AE	—	LLINMØ	Longitud de línea en SCREEN Ø. Valor por defecto=39
F3AF	—	LLINM1	Longitud de línea en SCREEN 1. Valor por defecto=31
F3BØ	—	LONLIN	Longitud de línea. Esta dir. está modificada por el valor de WIDTH
F3B1	—	LONPAG	Longitud de la página, es decir, el número de líneas vistas por pantalla. Su valor es 24 por defecto
F3B2	—	TAB	Número de caracteres para TAB. Valor por defecto=14
F3B3	F3B4	TNPØ	Dirección de la Tabla de Nombres de Patrones (TNP) en SCREEN Ø
F3B5	F3B6	TCØ	Dirección de la Tabla de Colores (TC) en SCREEN Ø
F3B7	F3B8	TGPØ	Dirección de la Tabla Generadora de Patrones (TGP) en SCREEN Ø
FEB9	F3BA	TASØ	Dirección de la Tabla de los Atributos del Sprite (TAS) en SCREEN Ø
F3BB	F3BC	TGSØ	Dirección de la Tabla Generadora de Sprites (TGS) en SCREEN Ø
F3BD	F3BE	TNP1	Dirección de la TNP en SCREEN 1
F3BF	F3CØ	TC1	Dirección de la TC en SCREEN 1
F3C1	F3C2	TGP1	Dirección de la TGP en SCREEN 1
F3C3	F3C4	TAS11	Dirección de la TAS en SCREEN 1
F3C5	F3C6	TGS1	Dirección de la TGS en SCREEN 1
F3C7	F3C8	TNP2	Dirección de la TNP en SCREEN 2
F3C9	F3CA	TC2	Dirección de la TC en SCREEN 2
F3CB	F3CC	TGP2	Dirección de la TGP en SCREEN 2
F3CD	F3CE	TAS2	Dirección de la TAS en SCREEN 2
F3CF	F3DØ	TGS2	Dirección de la TGS en SCREEN 2
F3D1	F3D2	TNP3	Dirección de la TNP en SCREEN 3
F3D3	F3D4	TC3	Dirección de la TC en SCREEN 3
F3D5	F3D6	TGP3	Dirección de la TGP en SCREEN 3
F3D7	F3D8	TAS	Dirección de la TAS en SCREEN 3
F3D9	F3DA	TGS	Dirección de la TGS en SCREEN 3
F3DB	—	KEYCLK	Efecto sonoro de CLICK al pulsar una tecla. Valor de 1 = CLICK. Valor de Ø para ningún efecto sonoro.
F3DC	—	CSRY	Posición de la coordenada vertical del cursor (y)
F3DD	—	CSRX	Posición de la coordenada horizontal del cursor (x)
F3DE	—	FPFLG	Teclas de función. Valor Ø=KEY OFF. Valor 1=KEY ON
F3DF	F3E6	VDPR	Contenido de los 8 registros del VDP en el orden Ø a 7
F3E7	—		Valor de Ø
F3E8	—		Valor de 255=&HFF
F3E9	—	FORCOL	Color del texto. Valor normal=15
F3EA	—	BAKCOL	Color de fondo. Valor normal=4
F3EB	—	BDRCOL	Color de borde. Valor normal=4
F3EC	F3EE		Salto a la posición &HØØØØ C3 ØØ ØØ (JP ØØØØH)
F3EF	F3F1		Salto a la posición &HØØØØ C3 ØØ ØØ (JP ØØØØH)
F3F2	—		Octeto de atributo
F3F3	F3F4	QUEADR	Dirección de la tabla de espera
F3F5	—		Valor de 255=&HFF

F3F6	—		Sincronización de la exploración de las teclas
F3F7	—		Valor de 8Ø=&H5Ø
F3F8	F3F9	PUTKBU	Dirección del byte a escribir en el buffer del teclado
F3FA	F3FB	GETKBU	Dirección del byte a leer en el buffer del teclado
F3FC	F4ØE	CASATR	Estos 2Ø bytes constituyen los parámetros de entrada/salida utilizados por el cassette
F4ØF	F413	RSNXTP	Puntero para la instrucción RESUME NEXT
F414	—	ERRNUM	Contiene el último código de error
F415	—	LPOS	Contiene la posición de la cabeza de la impresora
F416	—	LPTFLG	Flag de salida impresora. Valor de 1 para salida por impresora. Valor de Ø para salida por pantalla.
F417	—	IMPMSX	Valor de Ø para impresora MSX. Otros valores para impresoras no-MSX
F418	—	CARFLG	Si el valor es distinto de Ø, el carácter de salida no está codificado
F419	F41B	VAL	Utilizado por la función VAL
F41C	F41E	CURLIN	Posición del cursor en línea x
F41F	F55D	CRUBUF	Buffer para la codificación de una línea BASIC
F55E	F661	KBDBUF	Buffer para el teclado
F662	—	DIMFLG	Flag de la instrucción DIM
F663	—	STD	Flag que indica el tipo de variable presente en el acumulador (DAC)
F664	F665	OPTYP	Tipo de operador
F666	F667	CURCAR	Puntero de texto para GET CHR$
F668	F671	CODSAV	Guarda temporalmente el código de la instrucción
F672	F673	MEMSIZ	Parte superior de la memoria utilizable por el BASIC. Su valor se modifica por la instrucción CLEAR
F674	F675	STKADR	Dirección superior del puntero de pila (STACK POINTER)
F676	F677	TXT	Dirección de inicio del texto del programa en BASIC
F678	F6AØ		Almacenamiento de descripciones temporales, de cadenas, ...
F6A1	F6A2	FORPTR	puntero para la instrucción FOR
F6A3	F6A4	LLGRDX	Dirección de la última línea DATA
F6A5	—		Etiqueta para FOR y USR
F6A6	—		Etiqueta para INPUT y READ
F6A9	—	DIRMOD	Flag: modo de programa o modo directo
F6AA	—	AUTOFL	Flag: Ø = AUTO. 1=no modo AUTO
F6AB	F6AC	CLN	Número de línea siguiente utilizado por AUTO
F6AD	F6AE	ALINC	Valor del incremento entre dos líneas en modo AUTO
F6AF	F6BØ		Puntero para la instrucción RESUME
F6B1	F6B2	SPADR	Guarda la dirección de la pila para manipular un error
F6B5	F6B6	CURLIN	Contiene el número de línea
F6B7	F6B8		Puntero para la instrucción RESUME
F6B9	F6BA		Número de la línea en tratamiento de error
F6BB		ERRLFG	Flag: Valor 255 durante el proceso de error. Valor Ø después de RESUME
F6BE	F6BF	OLDLIN	Número de línea después de STOP o END
F6CØ	F6C1	OLDTXT	Dirección del último octeto ejecutado
F6C2	F6C3	VARTAB	Dirección de la tabla de las variables simples
F6C4	F6C5	VTBTAB	Dirección de la tabla de matrices
F6C6	F6C7	FSLAD	Dirección de inicio del espacio disponible
F6C8	F6C9	NCHPTR	Puntero que sigue al último carácter en curso de ejecución
F6CA	F7BB	VDLT	Tabla de la declaración de las variables. Compuesta de 26 bytes (uno


18 FC
CE FE
9C FF
CEFE
F8
C9
01 18 FC
CD CE FE
01 9C FF
CD CE FE
01 F6 FF
CD CE FE

			por cada letra del alfabeto)
F7BC	F7C3	TEMSWA	Zona temporal de almacenamiento para SWAP
F7C4	—	TROFLG	Flag: Valor ∅=TROFF. Valor 1=TRON
F7C5	—	BCDBUF	Inicio de la zona de trabajo del programa matemático
F7F6	F846	ACCUM	Acumulador matemático (DAC)
F847	F85E	ACCUM2	Acumulador secundario (ARG)
F857			Inicio de la zona de parámetros para la manipulación de los ficheros
F87F		FNCT	Contenido de las teclas de función (F1-F1∅)
F91F	F929	BASETB	Valor de las tablas del VDP
F92A		GENGRP	Zona de trabajo para el programa gráfico
F931		CIRCLE	Zona de trabajo para la instrucción CIRCLE
F949		PAINT	Zona de trabajo para la instrucción PAINT
F956		PLAY	Zona de trabajo para la instrucción PLAY
F975		VOICAQ	Dirección de las teclas musicales
FBB1			Valor distinto de ∅ si el BASIC está en ROM
FBCC		CURCOD	Código del cursor
FBCE			Etiquetas para ON KEY GOSUB
FBD8			Etiquetas para ON...GOSUB
FBDA		OLDKEY	Estado de la última tecla
FBE5		NEWKEY	Estado de la nueva tecla
FBF∅		KEYBUF	Buffer para el código de teclado
FC48	FC49	BOTTOM	Dirección de inicio de la memoria RAM
FC4A		HIMEM	Dirección final de la memoria RAM
CF9A		RTYCNT	Control de interrupción
FC9B		INTFLG	Flag de interrupción
FC9C	—	PADY	Valor vertical del joystick analógico
FC9D	—	PADX	Valor horizontal del joystick analógico
FCA∅	FCA1	INTVAL	Valor del intervalo para ON INTERVAL GOSUB
FCA2		INTCNT	Contador del intervalo
FCA6		GRPENT	Encabezamiento de carácter gráfico
FCA7			Contador de la secuencia ESCAPE
FCA8		INSFLG	Flag del modo de inserción
FCA9		CSRMOD	Flag del cursor ON/OFF
FCAA		CURCAR	Carácter del cursor
FCAB		CAPFLG	Flag de CAPS LOCK
FCAE		BASLOD	Flag de carga del programa BASIC
FCAF		SCRMOD	Modo de pantalla. Tipo de SCREEN
FCB∅		OLDMOD	Antiguo modo de pantalla
FCB2		PANCOL	Color de contorno para PAINT
FCB3	FCB4	GCSRX	Posición horizontal del cursor en modo gráfico
FCB5	FCB6	GCSRY	Posición vertical del cursor en modo gráfico
FCB7	FCB8	GRACX	Acumulador gráfico X
FCB9	FCBA	GRACY	Acumulador gráfico Y
FCBB	—	DRAWFG	Etiqueta para DRAW
FCBC	—	SCALE	Escala para DRAW
FCBD	—	ROT	Angulo para DRAW
FCBE			Flag de entrada/salida para código máquina (BLOAD/BSAVE)
FCBF	FCC∅		Dirección de ejecución de un programa cargado por BLOAD
FCC1		SLOTAR	Inicio de la zona de trabajo para la conmutación del cartucho
FD9A		HOOK	Inicio de la zona de enganches (HOOKS)


CHIP!
¿Y CUANDO DICES QUE ES SU CUMPLEAÑOS?
CREO QUE DENTRO DE QUINCE DÍAS...
¡NO!
SÍ, CHICO... ES TRISTE PERO ES ASÍ...
¡QUÉ FUERTE!
¡EEH! PASSA CON VOSOTROS?
¿SE OS HA ACABADO LA GARANTÍA O QUÉ?
OTRO QUE NO SE HA ENTERADO...
¿ENTERADO? ¿DE QUÉ ME TENGO QUE ENTERAR?
PUES QUE DENTRO DE QUINCE DÍAS ES EL CUMPLEAÑOS DEL CHAVAL...
O SEA, DE NUESTRO DUEÑO...
¡AH, PUES MUY BIEN, NO? ¿QUÉ LE REGALAMOS?
NO SEAS BURRO... ¿NO RECUERDAS EL CUMPLEAÑOS ANTERIOR? LE REGALARON VEINTE JUEGOS... VEINTE...
UNO DETRÁS DE OTRO
EL MISMO DÍA
SIN CANSARSE
ESTUVIMOS TRES DÍAS CON SUS TRES NOCHES SIN PARAR...
...CUANDO ÉL SE IBA AL COLE LE COGÍA EL RELEVO SU MADRE...
¿Y ESTE AÑO CUANTOS CREÉIS QUE SERÁN...?
PUES CONTANDO QUE HA HECHO NUEVOS AMIGOS YO CALCULO QUE TREINTA...
AAAAGRRRLLL...

```
540 PRINT "5";:PRINT
    "-PASO"
550 PRINT:PRINT "TECLEA UNA
    INSTRUCCION"
600 LET I$=INKEY$:IF I$=""
    THEN GOTO 600
610 IF I$<"1" OR I$>"5" THEN
    GOTO 600
620 I=VAL(I$):ON I GOSUB
    1000,2000,3000,4000,
    5000
700 IF A(M,2)<0 THEN GOTO
    7000
710 ER=ER+INT
    (RND(1)*1000)-200
720 IF INT(RND(1)*1600)
    -A(M,3)<0 THEN GOSUB
    900

740 A(M,1) =A(M,2) +A(M,
    3)*ER
750 COLOR 15,1,1:
    CLS
790 NEXT M
800 NEXT N
810 COLOR 15,1,1:
    CLS
820 PRINT "FIN DE
    JUEGO"
830 PRINT TAB(5);"ACTIVOS
    TOT.DE ";A$(1):PRINT
    TAB(11);"$";
    A(1,1)
840 IF NOP=2 THEN PRINT
    TAB(5);"ACTIVOS TOT. DE
    ";A$(2):PRINT
    TAB911);"$";A92,
    1)
850 PRINT TAB(2);"CUALQUIER
    TECLA PARA JUGAR DE
    NUEVO"
860 '
870 IF INKEY$="" THEN GOTO
    870
880 RUN
900 COLOR 15,1,1
905 JK=INT (RND(1)*
    100) + 50:IF JK>
    A(M,3) THEN JK=
    A(M,3)
910 LOCATE 8,9:PRINT " R O B
    O "
920 PRINT '' HAN SIDO
    ROBADOS ";JK;" KG DE
    ORO": A(M,3) =A(M,3)
    -JK:A(M,1) -A(M,1)
    -(JK*ER)
930 FOR X=1 TO 35:BEEP:NEXT
    X
940 COLOR 15,1,1:
    RETURN
```

ANALISIS DETALLADO DEL PROGRAMA

Para empezar, la línea 10 te pide que especifiques el número de jugadores y la línea 20 comprueba que la respuesta suministrada a la sentencia INPUT está dentro del margen permitido.

La línea 30 define p y nop con arreglo al número de jugadores.

En la línea 40 se dimensionan una serie de matrices juntamente con el precio del oro, er. La matriz a se utiliza para almacenar información acerca de los activos pertenecientes a cada uno de los jugadores y las minas; la matriz c se utiliza para almacenar información sobre las minas; la matriz a$ contiene los nombres de los jugadores y la r se utiliza para indicar si se han iniciado o no las labores de minería en la mina considerada por el jugador. La línea 50 inicializa los activos y el estado de la mina para ambos jugadores. Se asigna el valor 0 a r(1) y r(2) para indicar que todavía no se ha iniciado la minería en la primera mina en la que se van a hacer prospecciones. Otros valores asignados son los siguientes: a(1,1) y a(2,1) son los activos totales de cada jugador; a(1,2) y a(2,2) son los valores de efectivo de cada jugador; a(1,3) y a(2,3) son las cantidades de oro de cada jugador; a(1,4) y a(2,4) son los costes de la minería; a(1,5) y a(2,5) son los números de minas y finalmente a(1,6) y a(2,6) son las profundidades de cada mina. La línea 70 permite introducir el nombre de cada jugador.

El programa contiene un par de bucles FOR... NEXT, que empiezan en la línea 200 y terminan en las líneas 790 y 800. Estos son los bucles que definen el menú principal de opciones y la presentación de los ingresos de la compañía debidos a la minería, los costes de extracción, etc.

LAS VARIABLES N Y NOP

La variable n cuenta el número de pasadas realizadas por el jugador. La variable nop sirve para asegurarse de que ambos jugadores llegan hasta 30 pasadas. Más adelante el programa utiliza estas mismas variables para asegurarse de que se presentan los activos de ambos jugadores, etc.

En la línea 202 se definen los colores de la pantalla. La línea 210 presenta el título del juego: MINA DE ORO.

La línea 220 es la encargada de presentar el nombre o nombres de los jugadores. Cuando se elige la opción de dos jugadores, solamente se presenta el apellido.

Las líneas 230 a 300 presentan los valores de ACTIVOS TOTALES, CAJA EFECTIVO, CANTIDAD EN ORO, COSTES DE MINERIA, NUMERO DE MINAS, PROFUNDIDAD DE LA MINA Y PRECIO DEL ORO. Si hay dos personas jugando, se presentan los valores de ambos en los lugares apropiados, examinando los valores correspondientes a la variable nop.

La línea 400 presenta el nombre del jugador a quien le toca el turno en cada momento. Las líneas 500 a 540 le brindan al jugador las opciones Investigación y Desarrollo, Exploración e Informe, incremento de la profundidad de la mina en 200 metros. Precio del oro en dólares o pasar. La línea 550 invita al jugador a que teclee una instrucción.

Las líneas 600 a 620 utilizan la función INKEY$ para tener en cuenta la elección del jugador, comprobar que se trata de una elección válida y llamar a la subrutina encargada de su procesamiento.

La línea 700 examina si el valor total de los activos ha caído por debajo de cero, forzando un salto a la rutina de «final de juego» cuando así ocurre. En el próximo capítulo veremos la línea 7000 y siguientes. La línea 710 introduce fluctuaciones aleatorias en el precio del oro, por lo que tienes que



tener cuidado para vender tu oro en un momento en que su precio te permita hacer una operación favorable.

La línea 720 establece una comparación entre un número aleatorio y la cantidad de oro almacenada en tus cámaras acorazadas, al objeto de decidir si va a haber un robo o no; observa que son mayores las probabilidades de que ocurra un robo cuando tienes una gran cantidad de oro que cuando tienes una cantidad pequeña. La rutina de robo se extiende desde la línea 900 hasta la 940. La línea 905 elige la cantidad de oro que ha sido robada y la 920 se ocupa de presentar dicha cantidad en la pantalla.

La línea 740 calcula el valor total de los activos, sumando al valor de efectivo en caja el valor resultante del oro al precio vigente en cada momento. La línea 350 inicializa los colores de la pantalla, borrándola antes de que la instrucción NEXT envíe nuevamente el programa a la línea 200, dejándolo listo para la siguiente pasada.

Las líneas 810 a 840 constituyen la rutina de «juego terminado», que se utiliza cuando el activo total de uno de los dos jugadores ha caído por debajo de cero. La rutina presenta el estado financiero de ambos jugadores después de presentar el mensaje de FIN DE JUEGO.

Finalmente las líneas 850 a 880 corresponden a una rutina de ¿quieres jugar otra vez?

PARA OBTENER UN JUEGO MAS ADICTIVO

En el próximo capítulo veremos una serie de subrutinas que hacen que el juego resulte realmente adictivo. Veremos una rutina que permitirá reducir tus costes de minería por medio de la investigación y desarrollo, leer un informe relativo a las prospecciones realizadas en una mina, realizar excavaciones en la misma, escalonadas por etapas y cambiar tu oro por dólares.

Además veremos todos los datos que necesitarás para dibujar los gráficos que ilustran el estado de las minas de oro y el progreso realizado por las excavaciones.

IGUAL QUE EL REY MIDAS

- INVESTIGACION Y DESARROLLO DE NUEVOS METODOS EN MINERIA
- EXPLORACION DE NUEVAS MINAS
- EL INFORME SOBRE LA MINA
- PERFORACION DE LA MINA

Ha llegado el momento de que te hagas rico rápidamente. ¿Pero, has invertido en nuevas tecnologías antes de comenzar la exploración? ¿Cómo interpretas el resultado? ¿Y cuál es el mejor momento para vender? Tendrás que ser bastante perspicaz en la Mina de Oro.

Ya has visto en la primera parte de este juego cómo definir las distintas opciones que se ofrecen al jugador: Investigación y Desarrollo, Exploración e Informe, Aumento de la Profundidad de la Mina y Cambio de Oro por Dólares. Ahora puedes completar tu programa de la Mina de Oro con las subrutinas que manejan cada una de estas opciones.

Investigación y Desarrollo es la actividad que corresponde a la selección de la opción 1, Exploración Previa e Informe es la opción 2, Aumento de la Profundidad de la Mina es la opción 3 e Intercambio de Oro por Dólares es la opción 4. La opción 5 corresponde a pasar sin hacer nada, por lo que para ella no se requiere una subrutina completa. Las opciones 1, 2 y 4 introducen los elementos de aleatoriedad requeridos para que el juego se parezca de verdad al mundo real.

INVESTIGACION Y DESARROLLO

```
1ØØØ COLOR 15,1,1:CLS
1Ø1Ø LOCATE 4,3:PRINT
     "INVESTIGACION Y
     DESARROLLO":LOCATE 4,
     4:PRINT "(PARA
     DISMINUIR EL COSTE)"
1Ø2Ø LOCATE 6,7:PRINT
     "CUANTO QUIERE";
     TAB(5); "INVERTIR
     ($)":INPUT RD
1Ø5Ø A(M,4) =A(M,4)
     -INT(RD*.Ø5)-1
1Ø6Ø IF A(M,4)<Ø THEN A(M,
     4)=Ø
1Ø8Ø A(M,2)=A(M,2) -RD:A(M,
     1) =A(M,1)-RD
11ØØ LOCATE 3,13:PRINT "EL
     COSTE DE LA MINA";
     TAB(3);"SE HA REDUCIDO
     A $";INT(RD*.Ø5) A+1;"
     POR 2ØØM"
111Ø FOR Z=1 TO 3ØØ::NEXT Z
112Ø RETURN
```

La línea 1000 define los colores de la pantalla y borra ésta. A continuación la línea 1010 define la cabecera de pantalla antes de que la línea 1020 pregunte al jugador cuánto dinero quiere invertir en Investigación y Desarrollo; rd es la cantidad elegida.

La línea 1050 disminuye los costes de minería en una cantidad relacionada con el volumen de la inversión en Investigación y Desarrollo. En la línea 1060 se comprueba que los costes de minería no se hacen negativos. La línea 1080 ajusta los activos en metálico y totales para tener en cuenta la cantidad invertida en I + D.

La cantidad en que se ven reducidos los costes de minería se presenta en la línea 1100. La línea 1110 contiene un bucle FOR ... NEXT para introducir un pequeño retardo antes de que finalice la subrutina.

EXPLORACION E INFORME

```
2ØØØ COLOR 15,1,1:CLS
2Ø3Ø R(M)=Ø:C(M,1)=INT(RN
     D(1)*9Ø)+1Ø:C(M,2)=IN
     T((RND(1)*2ØØ:C(M,3)=I
     NT(RND(1)*2ØØ+1:LL=I
     NT(RND(1)*3)-1
2Ø5Ø C(M,4) =C(M,2) +LL*2ØØ
2Ø7Ø C(,5)=Ø: KK=INT
     (RND(1)*1ØØ):IF KK<C(M,
     1) THEN C(M,5)=1
2Ø8Ø LOCATE 6,2:PRINT
     "INFORME CIENTIFICO
     ":LOCATE 2,5:PRINT
     "POSIB. ENCONTRAR
     ORO= ";C(M,1);
     "%":LOCATE 2,7:PRINT
     "PROFUNDIDAD
     ESTIMADA= ";C(M,2);
     "M":LOCATE 2,9: PRINT
     "CANTIDAD ESTIMADA=
     ";C(M,3);"KG."
21ØØ Z=INT (RND(1)*15ØØØØ!):
     A(M,2)=A(M,2) -Z:A(M,1)
     =A(M,1)-Z
211Ø LOCATE Ø,12:PRINT "SE
     INICIAN EXCAVACIONES?
     (S/N)"
212Ø R$=INKEY$:IF R$=""
     THEN GOTO 212Ø
213Ø IF R$="S" THEN LET A(M,
     6)=Ø:R(M) = 1:GOTO
     3ØØØ
25ØØ RETURN
```

En la línea 2000 se borra la pantalla y se cambian sus colores. La línea 2030 pone a cero r(m) para indicar que todavía no ha empezado la excavación. En esta línea también se determina la probabilidad de encontrar oro, el valor esperado de la profundidad y la cantidad esperada. La variable ll es un número aleatorio comprendido entre 1 y −1, el cual se utiliza en la línea siguiente a fin de determinar la profundidad real del oro; recuerda pues que el valor de C(M,2) es precisamente el valor esperado de la profundidad.

La línea 2050 asigna a C(M,4) un valor igual a C(M,2) más o menos 200 metros (200 veces ll). Seguidamente la línea 2070 decide si la mina contiene realmente algo de oro. Se le asigna a C(M,5) el valor 0 para indicar que no hay oro. KK es un número aleatorio comprendido entre 0 y 99. KK se compara con la probabilidad de encontrar

oro, si KK es menor, entonces C(M, 5) se pone a uno para indicar que hay oro en la mina.

La línea 2080 presenta al jugador el informe científico de la mina. Aunque al jugador se le dice la probabilidad de que encuentre oro y la profundidad más probable, el que esto ocurra realmente depende de varios factores aleatorios. En consecuencia deberás utilizar tu propio juicio para determinar si la inversión merece la pena o no.

Ocupémonos ahora de las malas noticias: el informe hay que pagarlo. Es imposible predecir lo que costará, pero puede ser un valor comprendido entre nada y 150.000 dólares; éste es el valor de Z elegido en la línea 2100. El coste de la exploración y el informe hay que deducirlo de la disponibilidad en efectivo, y esta deducción aparecerá también en los activos totales.

En este momento al jugador se le ofrece la posibilidad de iniciar las excavaciones. En la línea 2110 se hace la pregunta ¿SE INICIAN LAS EXCAVACIONES? Si la respuesta es sí, el programa salta a la rutina de minería que empieza en la línea 3000.

EXCAVACION

```
3ØØØ COLOR 15,1,1:CLS
3Ø1Ø IF R(M)=Ø THEN LOCATE
     2,9:PRINT "NO HAS
     EXPLORADO
     TODAVIA":FOR Z=1 TO
     1Ø:BEEP:NEXT Z:RETURN
3Ø2Ø COLOR 15,1,1:CLS
3Ø22 PRINT TAB(14);CHR$(14
     7);CHR$(148);CHR$(149
     );TAB(14);CHR$(15Ø);CH
     R$(151);CHR$(152);CHR
     $(153);TAB(13);CHR$(1
     54);CHR$(155);CHR$(15
     6);CHR$(157);CHR$(158
     );TAB(31);CHR$(32)
3Ø25 FOR Z=1 TO 32:PRINT
     CHR$(144);:NEXT Z
3Ø6Ø LOCATE Ø,4:FOR Z=1ØØ
     TO 14ØØ STEP
     1ØØ:LOCATE Ø,3+Z/1ØØØ:
     PRINT Z :NEXT Z
3Ø9Ø A(M,2)=A(M,2) -A(M,
     4):A(M,1) =A(M,1)-A(M,
     4): A(M,6)=A(M,6)
     +2ØØ:FOR XX=1 TO
     2ØØ:NEXT XX
31ØØ LOCATE 15,3: PRINT
     CHR$(146):FOR F=4 TO
     (A(M,6)/1ØØ)+3:LOCATE
     15,F:PRINT
     CHR$(145):FOR W=1 TO
     1Ø:BEEP:NEXT W:NEXT F
312Ø IF A(M,6)=C(M,4) AND
     C(M,5)=1 THEN GOTO
     35ØØ
313Ø LOCATE 2,6:PRINT "NO
     ENCONTRASTE ORO
     TODAVIA":IF A(M,6)=C(M,
     2)+2ØØ THEN LOCATE Ø,
     18:PRINT "ESTA MINA NO
     TIENE ORO.PRUEBA CON
     OTRA.":FOR Z=1 TO
     1Ø:BEEP:NEXT Z:A(M,
     6)=Ø:R(M)=Ø
314Ø FOR XX=1 TO 3ØØ:NEXT
     XX
33ØØ RETURN
35ØØ LOCATE 12,F:PRINT
     "ORO":FOR Z=-2Ø TO
     5Ø:BEEP:NEXT Z:FOR
     XX=1 TO 5ØØ:NEXT XX
355Ø A(M,5)=A(M,5) +1:A(M,
     3)=A(M,3) +C(M,3):A(M,
     1) =A(M,1)+(A(M,3)*ER) :
     A(M,Y)=Ø:R(M)= Ø:GOTO
     33ØØ
```

Puedes acceder a esta rutina desde dos sitios del programa. Como ya has podido ver, se te ofrece la opción de iniciar las excavaciones desde la rutina de Exploración e Informe. Pero también se utiliza cuando optas por aumentar la profundidad de la mina en 200 metros, eligiendo el número 3 de la lista de opciones.

Como de costumbre, la primera línea de la rutina se limita a borrar la pantalla y a definir sus colores. La línea 3010 comprueba que se ha completado la fase de exploración para que pueda iniciarse la fase de excavaciones. La línea 3020 vuelve a ocuparse de nuevo de la pantalla, dejándola lista para nuevas presentaciones.

Las líneas 3022 a 3090 se ocupan de los gráficos que muestran en pantalla la mina de oro. La línea 3100 ilustra la excavación y genera algunos efectos sonoros.

La línea 3120 examina si la excavación ha llegado al nivel donde se encuentra el oro y si hay oro en la mina

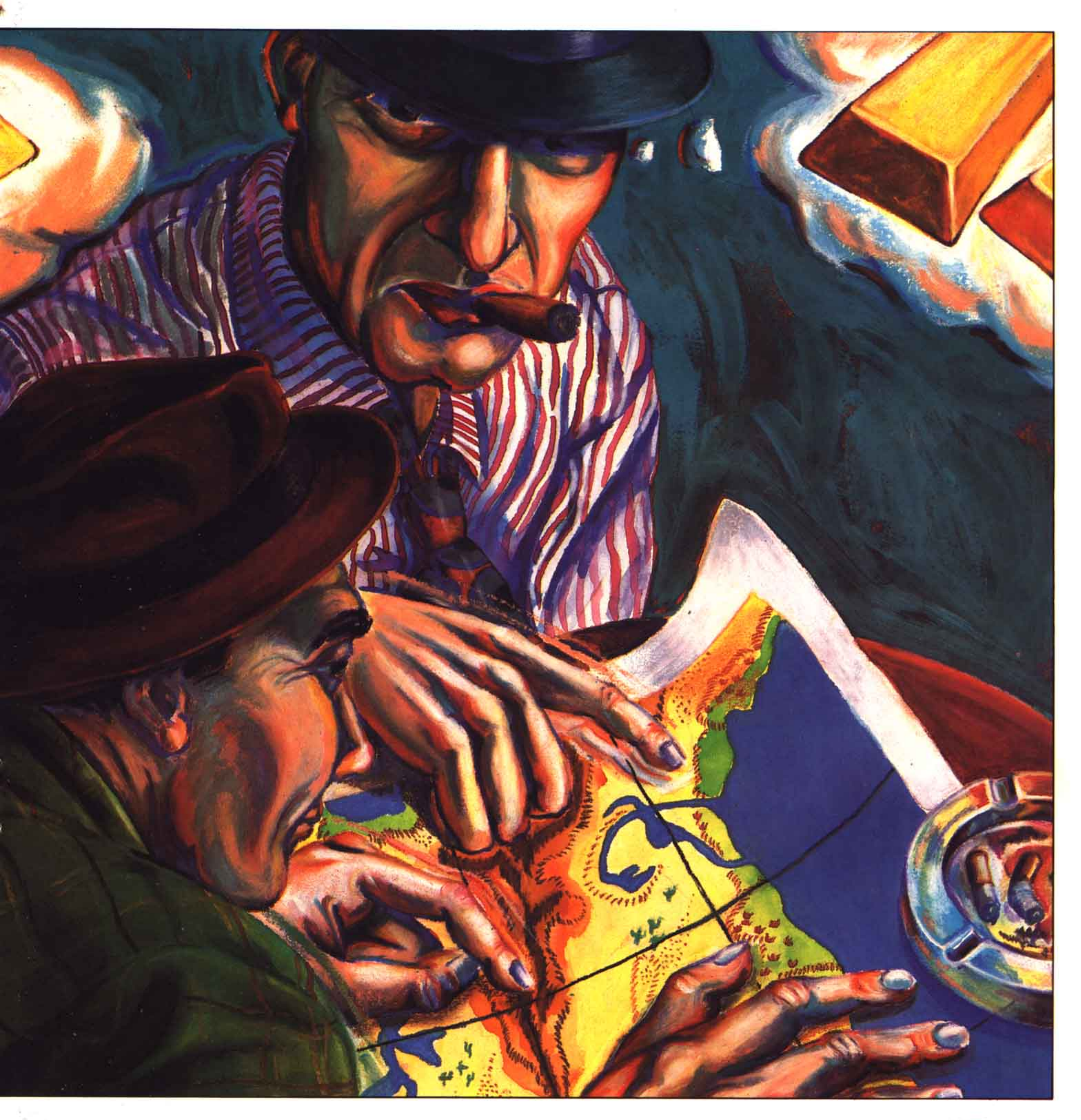


(podría ser que se llegara al nivel esperado para el oro y ocurriese que la mina no contiene absolutamente nada). Si se llega hasta el oro, el programa salta a la línea 3500, que informa al jugador de que se ha encontrado. La línea 3550 ajusta ahora el valor de los activos del jugador, con arreglo al valor del oro encontrado.

Si no hay oro, el programa continúa hasta la línea 3130. Si la excavación ha sobrepasado la altura esperada para el oro en 200 metros, el jugador es informado de que la mina no lo contiene. Si la excavación aún no ha llegado tan lejos, el jugador recibe el mensaje siguiente: NO HAY ORO TODAVIA.

EL PRECIO DEL ORO

```
4ØØØ COLOR 15,1,1:
     CLS
```



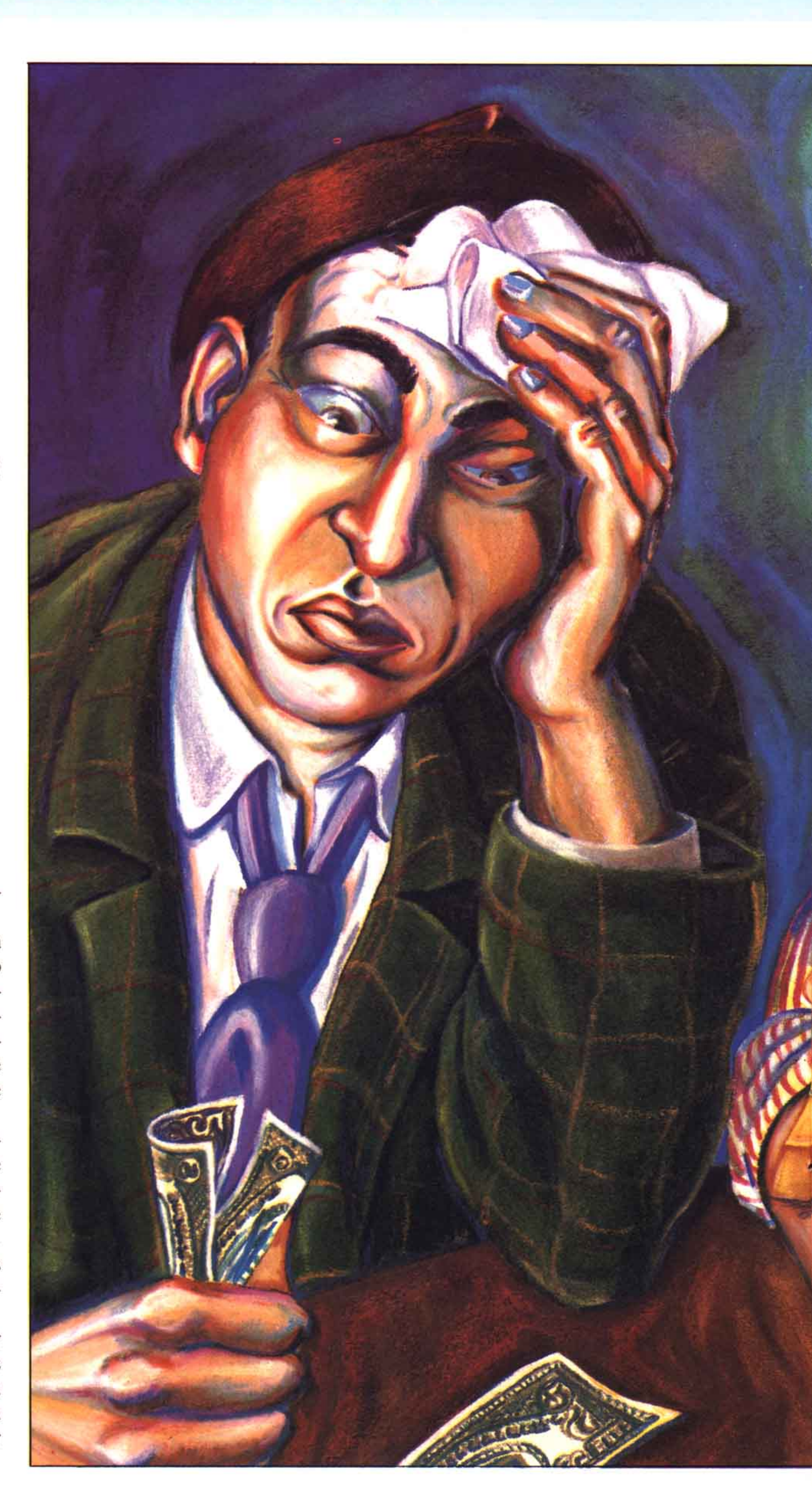
```
4Ø2Ø LOCATE 7,2:PRINT
     "AGENCIA DE CAMBIO
     ":LOCATE Ø,6:PRINT "EL
     TIPO DE CAMBIO ACTUAL
     ES:-":LOCATE 5,8:PRINT
     "1 KG DE ORO= $";
     ER:LOCATE 2,12:PRINT
     "CUANTOS KG QUIERES
     CAMBIAR":INPUT
     NTE
4Ø7Ø IF NTE>A(M,3) THEN
     LOCATE Ø,16:PRINT "NO
     HAY TANTO
     ORO"
4Ø8Ø NTE=INT(NTE)
4Ø9Ø IF NTE>A(M,3) OR NTE<Ø
     THEN GOTO
     4Ø2Ø
4Ø95 LOCATE Ø,16:PRINT
     CHR$(32); TAB(31);
     CHR$(32)
41ØØ A(M,3)=A(M,3)
     -NTE:A(M,2)=A(M,2)
     +(NTE*ER):A(M,1) =A(M,
     1)+(NTE*ER)
413Ø LOCATE 1,16:PRINT "KG
     CAMBIADOS POR $";
     NTE*ER:FOR XX=1 TO
     3ØØ:RETURN
5ØØØ RETURN
```

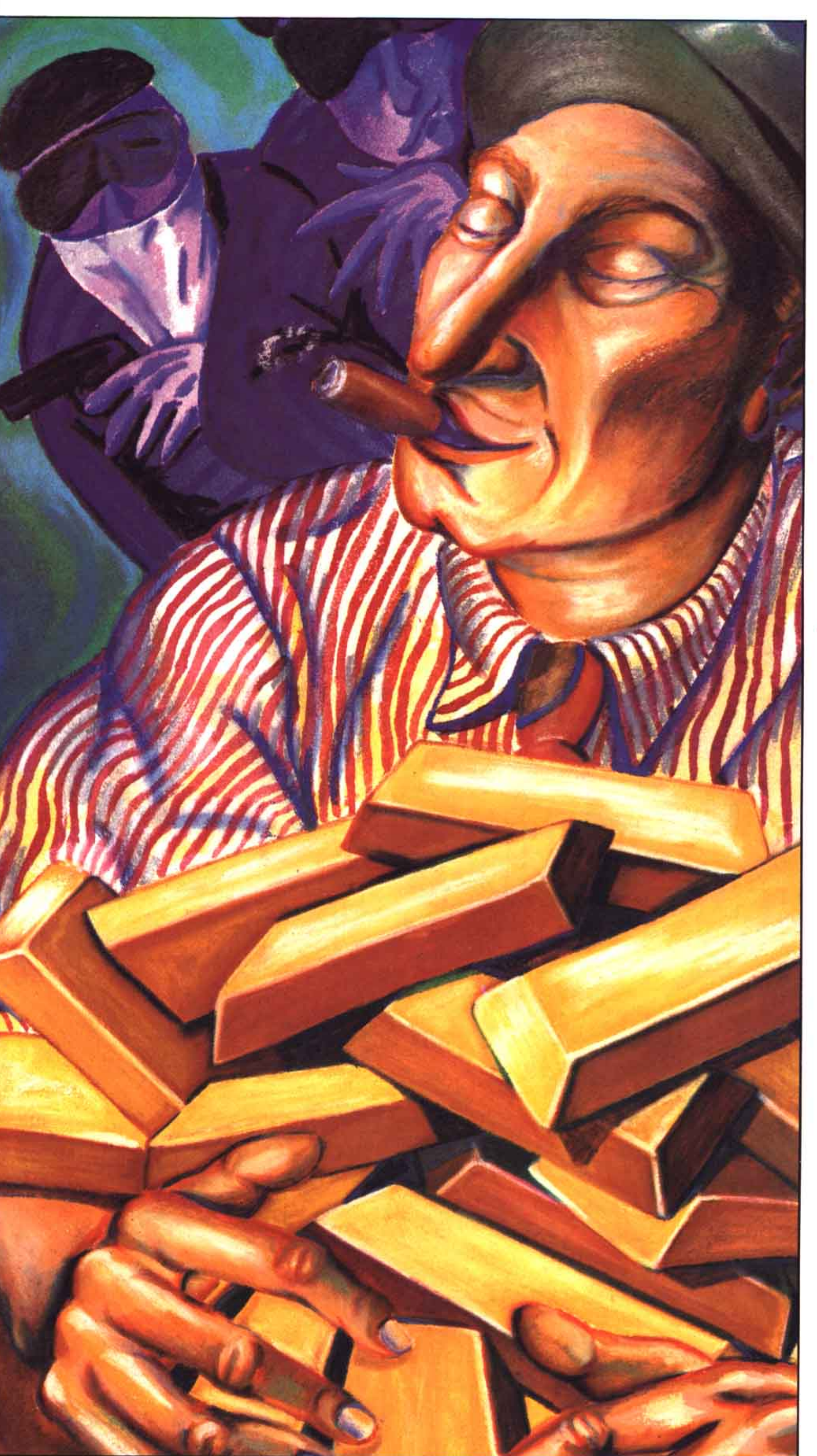
La línea 4000 configura el programa.

La línea 4020 presenta el título en la pantalla, el valor de mercado del oro y los mensajes para seleccionar el número de kilogramos que hay que vender. La línea 4070 comprueba si posees suficiente oro. La línea 4080 sirve para asegurarse de que la cantidad de oro vendido es un número entero.

La línea 4090 vuelve a enviar el programa al punto de presentación de mensajes si la cantidad que se pretende vender supera a la cantidad de oro que se posee,o es menor que cero. La línea 4100 modifica el valor de los activos totales con arreglo a la cantidad de oro vendido.

Esta subrutina informa al jugador de cuánto oro se ha vendido y cuántos dólares se han recibido a cambio, cosa que se hace en la línea 4130. La línea 5000 corresponde a la opción de pasar sin hacer nada.





LOS TOQUES FINALES

```
1 FORN=32ØØ 3312+7:READ
  A♁OKE N,A:NEXT
  N
7ØØØ  COLOR 15,1,1:
      CLS
7Ø1Ø  LOCATE 12,9:PRINT
      A$(M):LOCATE 8,1Ø:PRINT
      "HA HECHO
      BANCARROTA":LOCATE 1,
      2Ø:PRINT "PULSA UNA
      TECLA PARA JUGAR OTRA
      VEZ"
7Ø3Ø  DEFUSR=&H9F:
      X=USR(Ø): RUN
      5
8ØØØ  DATA 255,85,17Ø,Ø,Ø,Ø,
      Ø,Ø,62,28,56,126,28,62,
      12Ø,28
8Ø1Ø  DATA 255,255,62,126,
      127,6Ø,124,126,Ø,Ø,Ø,Ø,
      1,1,1,1
802Ø  DATA 7,29,49,45,255,
      255,91,126,128,96,48,
      8Ø,152,14Ø,252,
      138
8Ø3Ø  DATA 1,1,1,49,49,49,49,
      255,122,187,62,95,153,
      255,153,126
8Ø4Ø  DATA 2Ø9,177,224,128,
      128,128,128,128,Ø,Ø,
      128,128,64,32,32,
      16
8Ø5Ø  DATA 1,3,7,7,4,4,7,7,
      255,255,255,255,149,
      149,159,159
8Ø6Ø  DATA 24,126,153,255,
      126,153,126,219,128,
      192,224,24Ø,249,168,
      248,255
8Ø7Ø  DATA 16,8,8,4,14,31,31,
      255
```

Las líneas 7000 a 7030 contienen una rutina para jugar otra vez el juego.

Las líneas 8000 a 8070 contienen los DATA para los GRAFICOS.

¡Ahora ya puedes amasar tu inmensa fortuna y comprar todas esas cosas que siempre te has prometido adquirir un día!

DOMINANDO EL TABLERO

- CONSEJOS Y TRUCOS
- LA PRIMERA PANTALLA
- DEFINIENDO EL TABLERO Y LAS PIEZAS
- EL JUEGO EN MOVIMIENTO

Se levanta el telón para la presentación del Juego OTELO. Programa este sencillo juego de estrategia y de engaño y desafía a tu ordenador. Pero, ¡cuidado!, no es tan sencillo como parece.

OTELO es un juego de estrategia que se juega sobre un tablero de ocho por ocho casillas —un tablero de ajedrez o de damas puede servirnos—. Las reglas son muy simples y el juego cuenta con varios trucos.

El objetivo consiste en capturar o *comer* el mayor número de fichas posibles a tu oponente. Cada jugador ha de colocar una ficha en el tablero hasta llenarlo por completo. Cada jugador comienza con dos fichas y ha de intentar capturar las del jugador contrario *rodeándoselas*. Ello se logra colocando una ficha extra al final de una fila de fichas, de manera que el adversario se vea acorralado por tus fichas. Todas las fichas adversarias que hayas acorralado serán reemplazadas ahora por fichas tuyas.

El número de puntos será simplemente el número de fichas de cada jugador que haya sobre el tablero durante cada jugada. El ganador será aquel que consiga tener el mayor número de piezas sobre el tablero cuando éste esté lleno.

En esta versión, tú juegas contra el ordenador, el cual también muestra en pantalla el tablero y lleva la puntuación.

CONSEJOS Y TRUCOS

Al igual que en cualquier otro juego de estrategia, éste también cuenta con algunos trucos que pueden serte de gran ayuda. Si ésta es la primera vez que juegas al OTELO, los siguientes consejos te serán muy útiles.

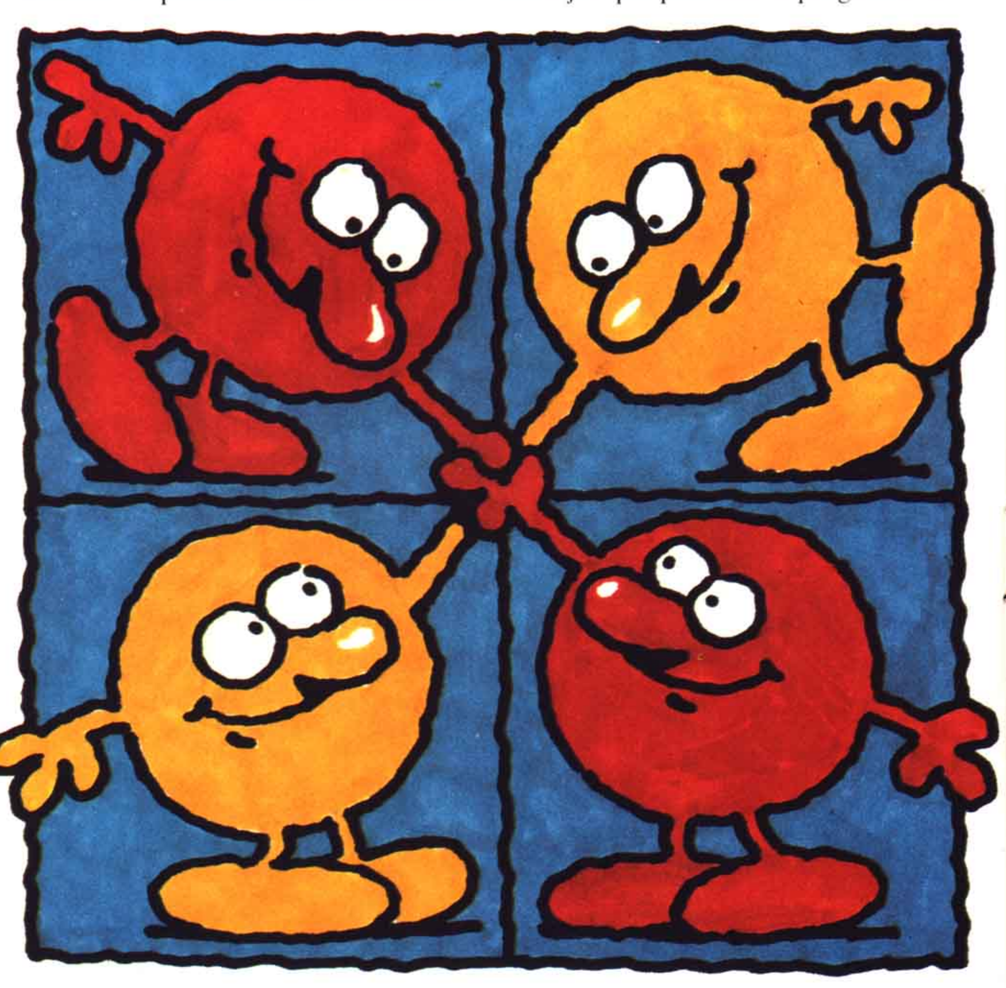
Las fichas de los extremos son muy valiosas, pues no pueden ser recuperadas una vez han sido capturadas —la razón de ello es debido a que estas fichas no pueden ser acorraladas como las de otras posiciones del tablero—. Como consecuencia, pueden ser muy significativas para ganar, y es muy importante capturar las esquinas, incluso sacrificando un movimiento que podría habernos proporcionado una mayor puntuación. Asimismo, también son intocables cada una de las fichas emplazadas junto a las fichas de las esquinas.

Puesto que una pieza puede enlazar más de una línea —arriba, abajo y en diagonal—, el movimiento más obvio no siempre será el mejor, mientras que en las últimas fases del juego, a menudo tú puedes enlazar dos o tres líneas añadiendo una sola ficha.

Debes pensar siempre con anticipación al adversario. Tal vez puedas conseguir engañar a tu oponente creando situaciones favorables para ti —para conquistar posiciones vitales— simulando una mala movida de tus fichas.

EL PROGRAMA

El programa juega el papel de tu oponente —con fichas negras—, y ya verás cómo un programa, comparativamente sencillo, es capaz de jugar con mucha destreza el desafiante juego OTELO. Una de las grandes ventajas que presenta el programa de



ARQUITECTURA DE LA CPU (II)

En el presente trabajo haremos una primera aproximación a los diversos bloques que componen la CPU Z80, para posteriormente hacer un análisis funcional de la misma como conjunto y ver así cómo realmente se pueden procesar datos con ella, supuestamente conectada a una memoria de programa, a una memoria de datos y a unos circuitos de Entrada/Salida.

El Z80 es un Microprocesador de 8 bits, con una capacidad de direccionamiento de memoria de 64K, que a diferencia del 8085, no lleva el Bus multiplexado, sino que utiliza pines completamente dedicados a funciones específicas, como son el Bus de Direcciones y el Bus de Datos.

Para una mejor comprensión vamos a dividir la CPU en bloques, que examinaremos detenidamente:

- Unidad Aritmético-Lógica (ALU), y Registro de Desplazamiento (SHIFT).
- Registros de Lenguaje Máquina, Bancos y Flags.
- Registro de Dirección de Memoria (MAR).
- Registro de Datos de Memoria (MBR).
- Unidad de Control.
- Bus de Control.

LA UNIDAD ARITMETICO-LOGICA

La ALU, como su nombre indica, está capacitada para realizar operaciones aritméticas y lógicas.

Las operaciones aritméticas son las que se refieren a cálculos como pueden ser sumas, restas, etc., y se utilizan, entre otras aplicaciones, para incrementar o decrementar registros o posiciones de memoria.

En cuanto a las operaciones lógicas son las que se realizan directamente con los bits. Dichos cálculos a nivel de bit, se utilizan generalmente para operar con máscaras, calcular direcciones, bloquear partes de datos, posicionar los flags, etc.

La ALU consta genéricamente de tres entradas y dos salidas que están distribuidas de la siguiente forma:

- 2 entradas para dos operandos
- 1 entrada para el Código de Operación
- 1 salida para el resultado
- 1 salida para los Flags de Estado

Con la intención de que no se preste a equívoco, puntualizamos que toda entrada o salida representa un Bus.

Es decir, un grupo de bits. El hecho de significarlo con una flecha, es una forma gráfica de presentación.

EL REGISTRO SHIFT

Conectado a la salida del resultado de la ALU se encuentra el Registro de Desplazamiento SHIFT, mediante el cual los bits de resultado de la ALU, pueden moverse en la dirección que indiquemos.

Básicamente, con una información podemos hacer dos tipos de movimientos: desplazamientos y rotaciones.

- Desplazamientos: La información se desplaza a la derecha o a la izquierda, perdiéndose el bit que sale fuera del registro.
- Rotaciones: El bit que sale del registro al producirse la rotación se hace entrar de nuevo por el otro lado.

Estas operaciones se denominan lógicas cuando el bit que entra es un cero, mientras que si copia el bit de mayor peso sobre sí mismo al desplazar a la derecha se considera una operación aritmética, ya que como veremos más tarde mantiene el signo.

Otra distinción que podemos hacer en este tipo de operaciones es si se realizan o no, a través del *Carry*, pudiendo formar así un registro de desplazamiento o rotación de 9 bits.

Cuando describamos las operaciones que tiene definidas nuestra CPU a nivel de Lenguaje Máquina veremos las Instrucciones de Desplazamiento y Rotación que puede realizar, descubriendo cada una de ellas. Como anticipo podemos enumerar las siguientes:

- Desplazamiento lógico y aritmético a la derecha
- Desplazamiento lógico y aritmético a la izquierda
- Rotación a la derecha de 8 y 9 bits
- Rotación a la izquierda de 8 y 9 bits

Debido a que los datos a desplazar son el resultado de la operación efectuada por la ALU, cuando queramos efectuar un desplazamiento, dicha unidad sólo tendrá que poner en la salida los datos de la entrada que seleccionamos y realizar el desplazamiento.

Si, por el contrario, la operación a realizar afecta a la ALU pero no al registro de desplazamiento, éste tan sólo dejará en la salida la información presente en su entrada.

Podemos pensar también en funciones combinadas de operación ALU seguidas por un desplazamiento en el SHIFT, que darán origen a instrucciones muy complejas y potentes, que aumentarían la eficiencia de la CPU.

Dado que el registro de desplazamiento está directamente conectado con la salida de la ALU y comparte también la entrada de código de operación, el conjunto ALU y SHIFT, puede ser representado como un único bloque llamado ALU, sobreentendiendo que alberga en su interior el mencionado registro.

LOS REGISTROS DE LENGUAJE MAQUINA

Para realizar los cálculos que impone la ejecución de un programa, es preciso mantener diversos estados in-

termedios temporales, contadores, índices de pequeños bucles, etc. Si estos valores los guardamos en la memoria principal, estaremos continuamente accediendo a la misma.

Los accesos a la memoria principal son lentos comparados con los movimientos de información por el interior de la CPU, ya que intervienen los buses principales del sistema, toda la lógica de decodificación de memoria, y la parte más lenta, que es la memoria externa, ya que debe seleccionar la celdilla direccionada y después leer o escribir la información en ella. Ante la lentitud del proceso, dentro de la CPU se han dispuesto una pequeña cantidad de memorias, capaces de ser escritas o leídas muy rápidamente, sin tener que acceder a la memoria principal del sistema, externa a la CPU. Estas memorias se llaman *registros*.

Algunos de estos Registros son accesibles y modificables desde el Lenguaje Máquina. Existen otros que sirven para realizar funciones internas de la CPU, como por ejemplo, los registros que se encuentran en la entrada de la ALU y que mantienen los datos temporalmente mientras se realiza la operación, a los que llamamos Registros Temporales.

Dichos registros, al contrario de los anteriores, no son accesibles ni se pueden modificar desde el Lenguaje Máquina.

Todos los Registros están conectados mediante unos Buses Internos que no son accesibles desde fuera de la CPU, conectados uno al Bus A y el otro al B. Existe un tercero, que es el Bus C, el cual recoge el resultado de la ALU y lo devuelve a los registros.

Existen dos grupos de Registros de Lenguaje Máquina, los de 16 bits y los de 8 bits.

De 16 bits encontramos éstos:
— Registro PC (Program Counter)
— Registro SP (Stack Pointer)
— Indice X
— Indice Y

Los Registros Indice X e Indice Y, se utilizan normalmente como punteros a memoria para direccionamientos indexados, mientras que los Registros PC y SP, son fundamentales para el control del programa externo.

El registro PC (Program Counter) es el contador de programa. Su misión es direccionar la posición de memoria donde se encuentra la instrucción a ejecutar.

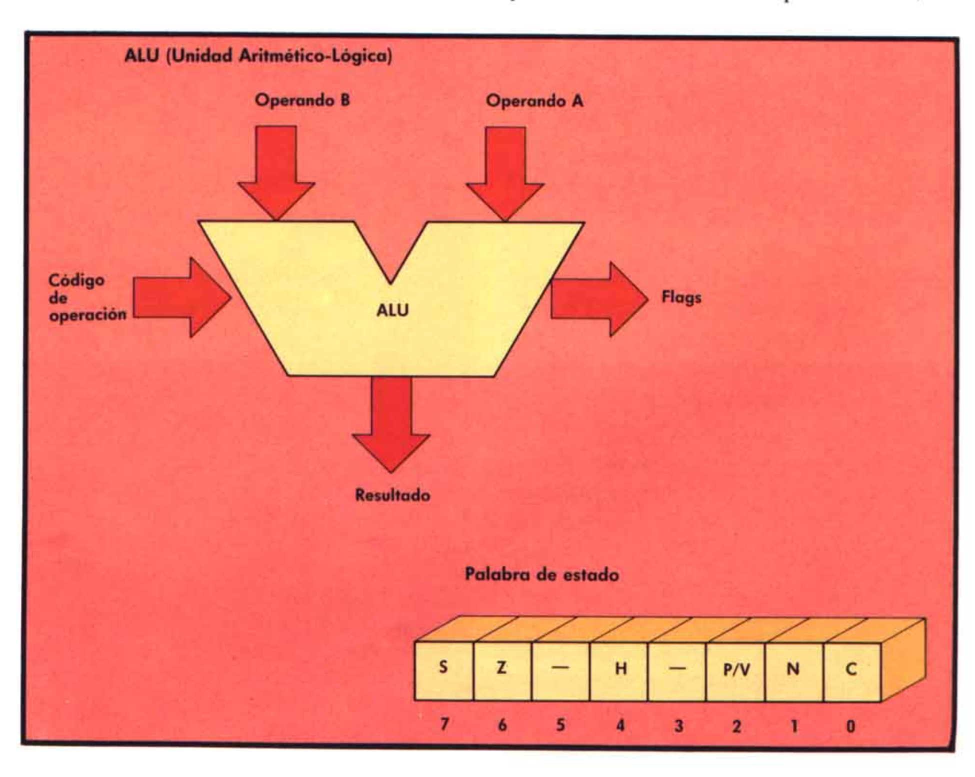
Se incrementa continuamente, siguiendo la ejecución del programa, de manera que siempre nos indicará en qué dirección está la siguiente instrucción de la memoria.

Cuando deseamos realizar un salto en el programa, cargaremos la nueva dirección en este registro, el cual hallará siempre la citada dirección en función del número de bytes que tenga la instrucción que esté ejecutando en ese momento.

El registro SP (Stack Pointer) es un puntero a Pila. La Pila es donde guardaremos la dirección de retorno en las llamadas a subrutinas, así como donde dejaremos los parámetros de llamada, o recogeremos los resultados de las subrutinas cuando empleemos el método de pasar variables, en los procesos de Subrutinas Recursivas. También podemos utilizarla para guardar el entorno de la CPU, cuando se produzca una interrupción en el sistema.

Los actos de guardar o recuperar datos de la Pila los haremos siempre de una forma ordenada, ya que en caso contrario, se perderían.

El Puntero de Stack tiene como finalidad señalar el último dato que ha entrado, de forma que si quisiéramos recuperar un dato anterior, debemos quitar primero los que hemos ido colocando encima, uno a uno por orden hasta llegar al deseado. A la vez que vamos cogiendo datos del Stack, el puntero también va desplazándose, de



manera que siempre señala el último dato que dejamos en la pila.

Al igual que los anteriores, estos registros son de 16 bits. Se utilizan para señalar posiciones de la memoria, pero a diferencia del SP, el puntero no incrementa ni decrementa automáticamente su posición, sino que permanece señalando hacia el lugar que le indiquemos. Esto permite operar con bloques de datos, independientemente de su localización física en memoria.

Siguiendo con los registros de Lenguaje Máquina, llegamos a los de 8 bits, que son los siguientes: B-C, D-E, H-L y A-F.

El registro de Lenguaje Máquina más importante es el Acumulador, registro operativo principal donde se guarda uno de los operandos de las operaciones aritméticas y lógicas, almacenando el resultado, después de haber realizado la operación.

Estos registros de 8 bits forman un bloque al que llamamos Banco.

A excepción del registro A (Acumulador, que es el Registro Operativo principal) y del registro F (Palabra de Estado), el resto de los registros de 8 bits podemos agruparlos de dos maneras, bien por separado, con lo que tendremos 6 registros de 8 bits, o bien por parejas, convirtiéndose en 3 registros de 16 bits.

Las dos formas de tratarlos son correctas, ya que existen instrucciones adecuadas para cada tratamiento.

Disponemos de dos Bancos de Registros, aunque no se pueden usar simultáneamente.

La existencia de los dos Bancos está justificada en casos, por ejemplo, de tener que realizar un cálculo rápido, del que no precisemos una constancia mientras trabajamos en un Banco.

Realizar esas operaciones en el mismo Banco, nos haría perder mucho tiempo debido a que tendríamos que guardar en la Pila los contenidos de los registros y luego recuperarlos. Disponemos de una única instrucción para el cambio de Bancos.

Aunque los dos Bancos sean iguales, siempre deberemos trabajar con ellos por separado.

Durante el tiempo que permanecemos operando en el segundo banco, los valores de los registros del otro, se mantienen. Pero deberemos tener en cuenta que, el uso continuado de esta instrucción de cambio, puede crearnos problemas.

Estos problemas podrían originarse si se produjeran interrupciones y éstas cambiasen de Banco, ya que podríamos perder el contenido de los registros de los dos Bancos.

Otra situación que puede darse es que si cambiamos constantemente de Banco, lleguemos a perder la noción de cuál sea el Banco en que nos hallamos, por lo que es aconsejable llevar un control riguroso del Banco donde estamos trabajando.

No pretendemos dar la imagen de que esta instrucción sólo puede acarrear problemas, sino que su existencia es importante y necesaria, pero consideramos que es imprescindible comentar los peligros que supone su utilización sin tener un conocimiento previo de los pros y los contras de la misma.

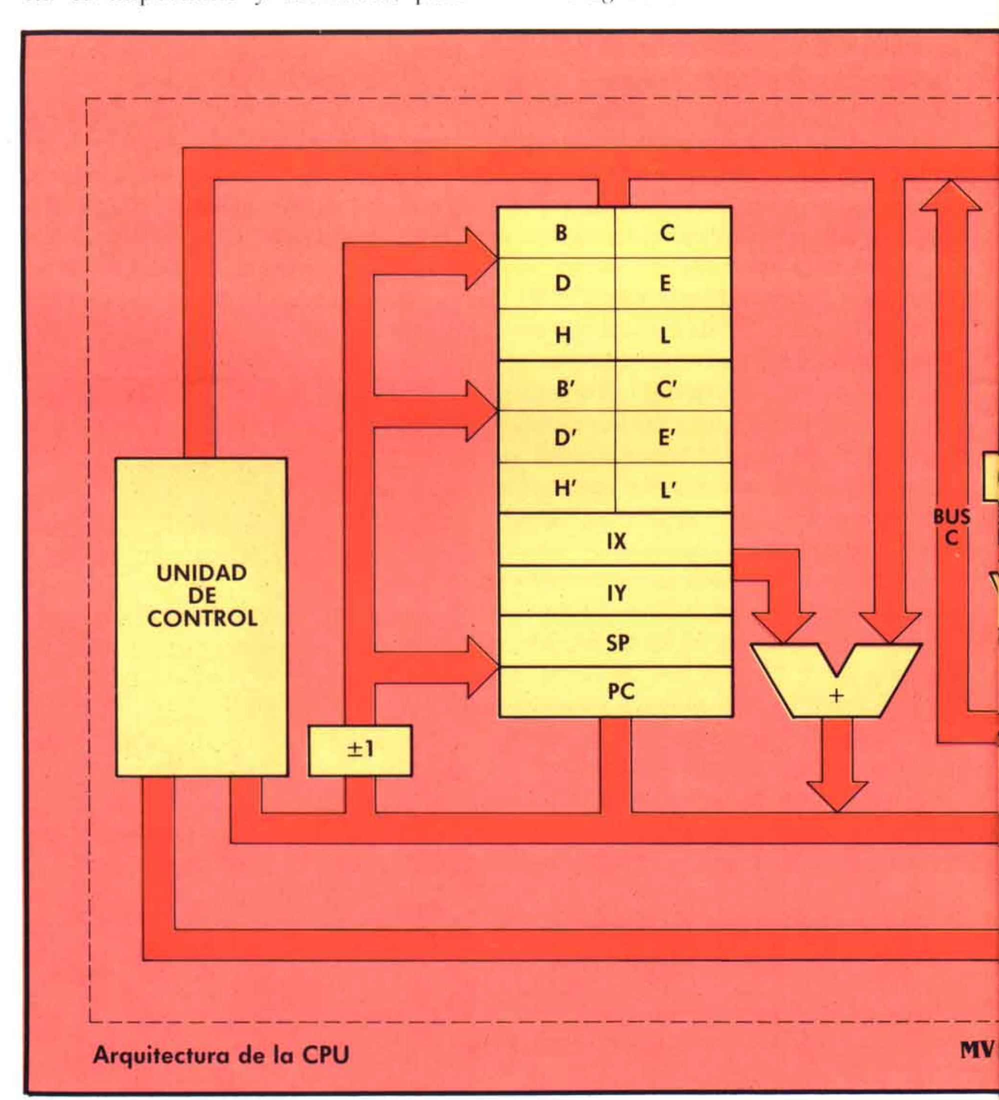
LOS FLAGS

Flag (palabra inglesa que podemos traducir como «banderola») es un bit que tiene como misión informar de los procesos de la ALU.

Dependiendo de estos indicadores habrá unas instrucciones en Lenguaje Máquina, que realizarán saltos, o no, según el Flag que les afecte. Estas instrucciones serán las llamadas de Salto Condicional.

La CPU genera dichos *Flags*, en función del resultado de las operaciones. Existen varios tipos de *Flags*:

— *Flag* de Cero



Arquitectura de la CPU

— *Flag* CY (Carry=Acarreo)
— Paridad y *Overflow*
— Signo (Minus)

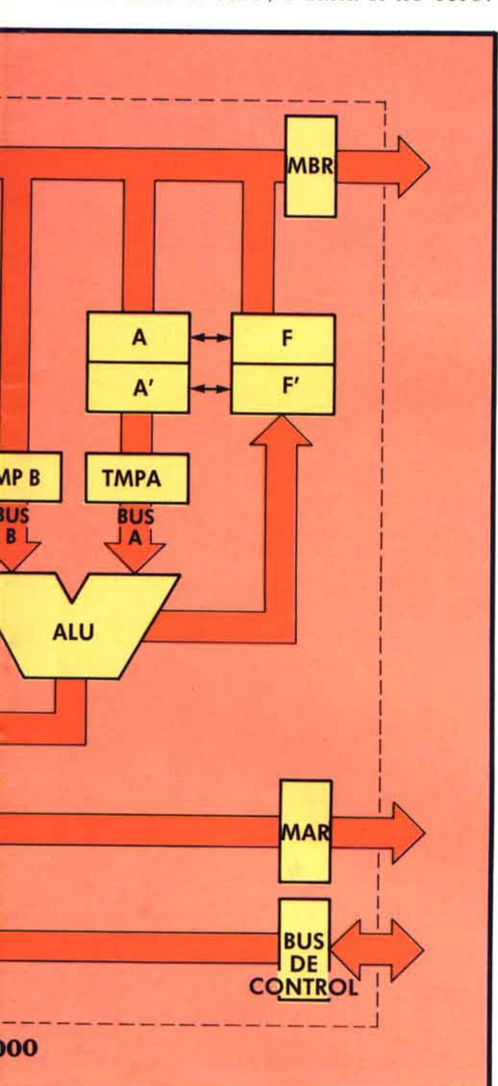
Flag de Cero

Es el resultado de la operación NOR de los bits del Acumulador. Su funcionamiento es el siguiente:

Si todos los bits del acumulador se encuentran a cero, la OR de todos los bits da 0, pero al invertir el resultado da 1, indicando que el acumulador está a 0.

Si por el contrario encuentra algún 1, el resultado de la OR será 1, pero al negarla será 0.

El empleo de este *Flag* es indispensable en programas donde sea necesario conocer el resultado de la operación, sirviéndonos de condicionante para el salto, dando pie a instrucciones como salta si cero, o salta si no cero.



Flag CY (Carry)

El *Carry* es un registro que se utiliza cuando se ha superado la capacidad del Acumulador, bien sea en una suma, una resta, o en cualquier operación aritmética.

Las operaciones lógicas no le afectan, porque siempre trabaja con los bits en sentido vertical.

En una operación donde el resultado sobrepase 255, el contador indicará tan sólo la parte baja del resultado.

En ese momento es cuando el Carry pasará a valer 1, indicando que se ha sobrepasado la capacidad del registro. Si deseamos llevar una cuenta de más de 8 bits, podremos ahora incrementar la parte alta del resultado.

Flag de Paridad y Overflow

Este *Flag* es de doble utilidad, dado que se puede utilizar como *Flag* de Paridad y como Overflow.

El Flag de Paridad tiene como misión mirar si el número de bits del resultado es par o impar, es decir, si en el resultado encuentra un bit a 1, el Flag se pondrá a 0 porque es impar, y viceversa, si encuentra dos bits a 1, se pondrá a 1 porque será par.

Este Flag no se utiliza normalmente, ya que se emplea en tests de memoria y de transmisiones de datos.

Los Tests de Memoria son realizados por ordenadores grandes, que guardan el bit de Paridad en la memoria y que lo utilizan para comprobar que dicha memoria está correctamente, pudiendo detectar sus fallos de lectura o escritura.

Como hemos citado este *flag* también se utiliza en comunicaciones, o sea, en trasvase de datos a distancias considerables, como podría ser de un ordenador a otro. Al llegar los datos se efectuaría un Test de Transmisión, donde se verifica la paridad del resultado. Si éste da 0, sabemos que es impar y por tanto confirmamos que la transmisión no ha sido correcta, debiendo efectuarla de nuevo.

El *Flag de Overflow* se utiliza cuando se trabaja con números con signo en complemento a dos, en operaciones aritméticas.

En el Z80, el *Flag de Overflow* indica que el complemento a dos de un número en el Acumulador tiene error, si excede de 127 o es menor de −128.

Flag de Signo (Minus)

Este *Flag* es el encargado de decir el signo de la operación. Su posición coincide con el bit más alto del resultado. Sólo se utiliza en operaciones con signo, ya que si no su empleo carece de sentido.

En total existen seis *Flags*, que están contenidos en un registro de 8 bits, por tanto hay dos bits que no se utilizan.

Dicho registro de Flags, también recibe el nombre de «Palabra de Estado», pues nos permite ver el estado en que se encuentran los *Flags* en cualquier momento.

Estos seis *Flags* podemos dividirlos en dos grupos, uno de cuatro flags y un segundo de dos.

En el primer grupo tenemos los que afectan a las instrucciones de Salto Condicional y a llamadas o retornos de subrutinas, también de forma condicional.

Estos Flags son los explicados anteriormente y se representan de la forma siguiente:

— «C» para el *Flag Carry*
— «Z» para el *Flag* de Cero
— «P o V» para el *Flag* de Paridad y *Overflow*
— «S» para el *Flag* de Signo (Minus)

El segundo grupo corresponde a *Flags* de uso exclusivamente interno de la CPU. No son accesibles ni siquiera desde el Lenguaje Máquina, ya que todavía están a un nivel más bajo de las Jerarquías del Computador, o sea, los encontramos a nivel eléctrico, de cableado de la CPU, y son necesarios para poder realizar instrucciones muy determinadas por la CPU.

Estos *Flags* son: el HALF CARRY y el *Flag* N.

El *Flag* HALF CARRY se produce cuando hay *Carry* desde los cuatro bits más bajos hasta el quinto bit.

El *Flag* N se utiliza para operaciones en BCD, en sustracciones, adiciones y también se emplea en la instrucción DAA (Ajuste decimal del Acumulador).

LOS MEJORES DE INPUT

JULIO 1987

MSX

PUESTO	TITULO	PORCENTAJE
1.º	SOCCER	21,1 %
2.º	GREEN BERET	18,4 %
3.º	NEMESIS	15,7 %
4.º	THE GOONIES	11,6 %
5.º	KNIGHT LORE	8,5 %
6.º	KNIGHTMARE	5,9 %
7.º	LIVINGSTONE	5,8 %
8.º	PROFANATION	4,5 %
9.º	HERO	4,4 %
10.º	BATMAN	4,1 %
		100,0 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «LOS MEJORES INPUT».

ELIGE TUS PROGRAMAS

Hemos pensado que es interesante disponer de un *ranking* que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en INPUT.

El resultado de las votaciones será publicado en cada número de INPUT.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: LOS MEJORES DE INPUT Aribau, 185. Planta 1. 08021 Barcelona — MSX N.º 14

1.º Título elegido ____________ Qué ordenador tienes ____________

2.º Título elegido ____________ Nombre ____________

3.º Título elegido ____________ 1.º Apellido ____________

Programa que te gustaría conseguir ____________ 2.º Apellido ____________

Fecha de nacimiento ____________ Teléfono ____________

Dirección ____________

Localidad ____________ Prov. ____________

TODO SOBRE

VAMPIRE KILLER (Y II)

Continuando la explicación referente al programa VAMPIRE KILLER, este mes os hacemos entrega de la segunda parte del mapa y del comentario correspondiente, hecho nivel a nivel.

El décimo nivel es uno de los difíciles. Una de las mayores dificultades a superar, son los grandes abismos que debemos pasar gracias a las placas móviles que se mueven por encima de ellos. Antes de aventurarte a superar el primer abismo, ve hacia tu izquierda y recoge una llave para abrir cofres; la otra es imposible de alcanzar por este camino. Una vez la tengas, y gracias a tu propia habilidad, desplázate hasta el cofre y ábrelo: se te recompensará con una cadena de combate. Continúa en línea recta y ve superando los diferentes obstáculos que se interponen entre tú y la llave. Sentimos deciros que el único truco existente en este nivel es vuestra destreza, pero os advertimos que para pasar por algunos pasadizos os será necesario agacharos.

El nivel número once es lineal, y toda la dificultad estriba en superar los ataques de los jorobados. Esquívalos mientras salten, y espera a que se arrastren por el suelo para atacarlos. En este nivel es muy importante conseguir el hacha de guerra; en el mapa te indicamos dónde se encuentra, pero recuerda llevar una llave para abrir el cofre que la contiene.

El decimosegundo nivel se caracteriza por su recorrido a través de pasadizos y por estar lleno de calaveras de dragón. Para destruir estas calaveras, atácalas por la espalda, y dispara una y otra vez hasta que desaparezcan. Por lo que respecta al laberinto de pasadizos, fíjate en el mapa para guiarte. Como en la Cámara del Mal te será necesaria un arma de largo alcance, si has perdido el hacha de guerra ve hasta la bruja que se encuentra en este nivel, y ella te venderá una estrella de combate. Una vez dentro de la Cámara del Mal sitúate en un rincón y dispara hasta destruir al «pequeño» Frankenstein.

El nivel número trece se ambienta en las celdas de castigo y las mazmorras del castillo, y seguro que para los supersticiosos se convertirá en un nivel fatal. Pero no te dejes intimidar por este tipo de creencias, te será fácil acceder al final de este nivel, aunque el camino sea largo y fatigoso. Por ello, cuando estés cansado y tu nivel de vida haya bajado, puedes hacer uso de la bola de energía que hay justo donde se te indica en el mapa. Como enemigos, destacan en este nivel los esqueletos rosas. Éstos son indestructibles y nuestros ataques sólo provocarán en ellos un momentáneo desfallecimiento; durante este lapso de tiempo debemos salir de su alcance. A los otros enemigos ya los conoces, y son los jorobados y los esqueletos normales, que te seguirán atormentando con sus huesos. El recorrido a realizar, hasta la llave y la puerta, tiene dos claras alternativas; las ventajas e inconvenientes de cada una te toca a ti descubrirlos.

El nivel número catorce es más sencillo, en cuanto a laberinto se refiere, que el nivel anterior, pero más difícil si nos fijamos en los enemigos, y sobre todo según el arma que lleves. Con las armas de corto alcance te será muy difícil destruir a tus nuevos enemigos lanzadores de hachas. Pero con las armas de largo alcance como mínimo nos podremos poner fuera de su línea de tiro para realizar nuestro ataque. De quien te seguirá siendo imposible librarte es de los esqueletos rosas. Para pasar este nivel recorre lo más rápido posible las ocho pantallas que lo forman.

AVANZANDO HACIA EL CONDE DRACULA

El nivel número quince es muy parecido a los dos anteriores, aunque por fin te has librado de los esqueletos rosas. En este nivel es muy importante hacerse con el reloj de arena; os indicamos en el mapa la situación del cofre que lo contiene. Para acceder a la llave deberás caerte por el agujero que hay delante de la puerta que lleva a la Cámara del Mal. Una vez te hayas dejado caer vuelve a tirarte, pero saltando hacia la izquierda. Sube otra vez y entra en la penúltima Cámara del Mal. Ésta es la más bella y difícil de todas, y para superarla deberás hacer uso inteligente de tu arma y el reloj de arena que anteriormente recogiste. No te precipites en tus ataques.

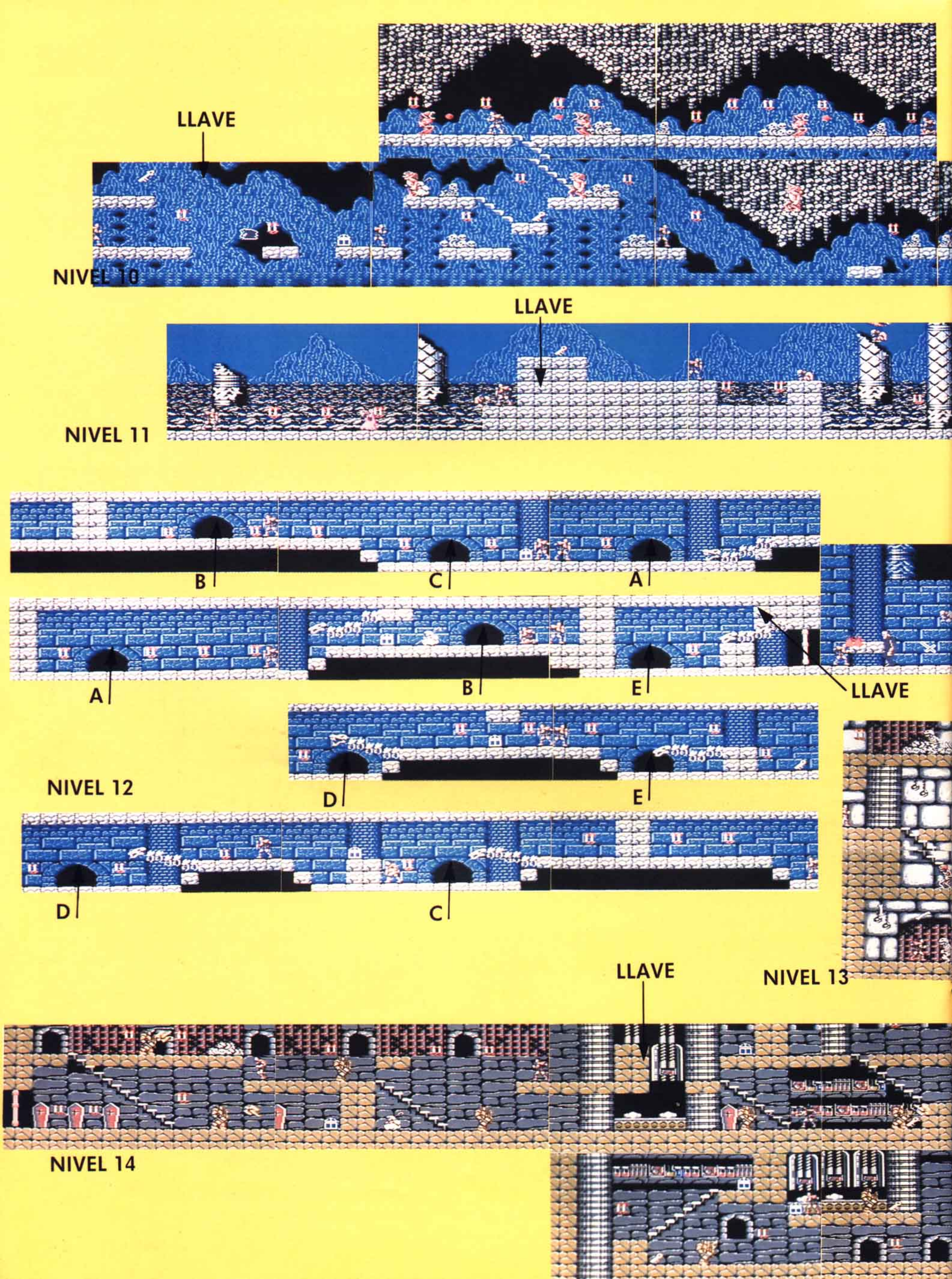
El nivel número dieciséis se ambienta en el exterior del castillo, y es uno de los niveles hechos con más picardía, ya que en él hay tres llaves. Si quieres arriesgarte a encontrar las llaves, ve por el camino inferior, pero si quieres salir rápidamente de este nivel, coge el camino de arriba y no pares por nada. Al final, delante de una puerta, encontrarás la llave.

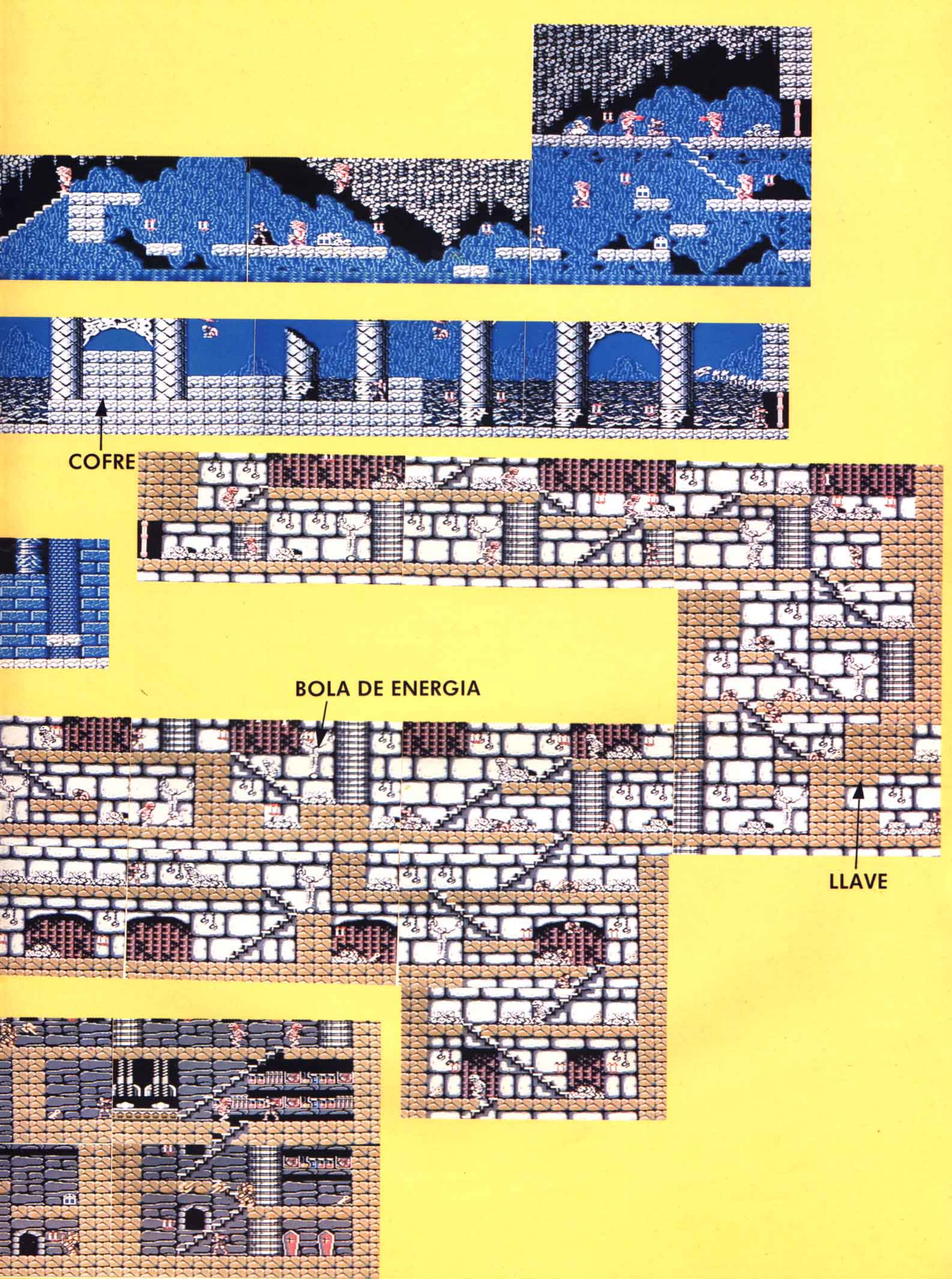
En el nivel número diecisiete, también hay más de una llave, pero la única que puedes alcanzar es la que te indicamos en el mapa. Y para ello deberás realizar las siguientes acciones:

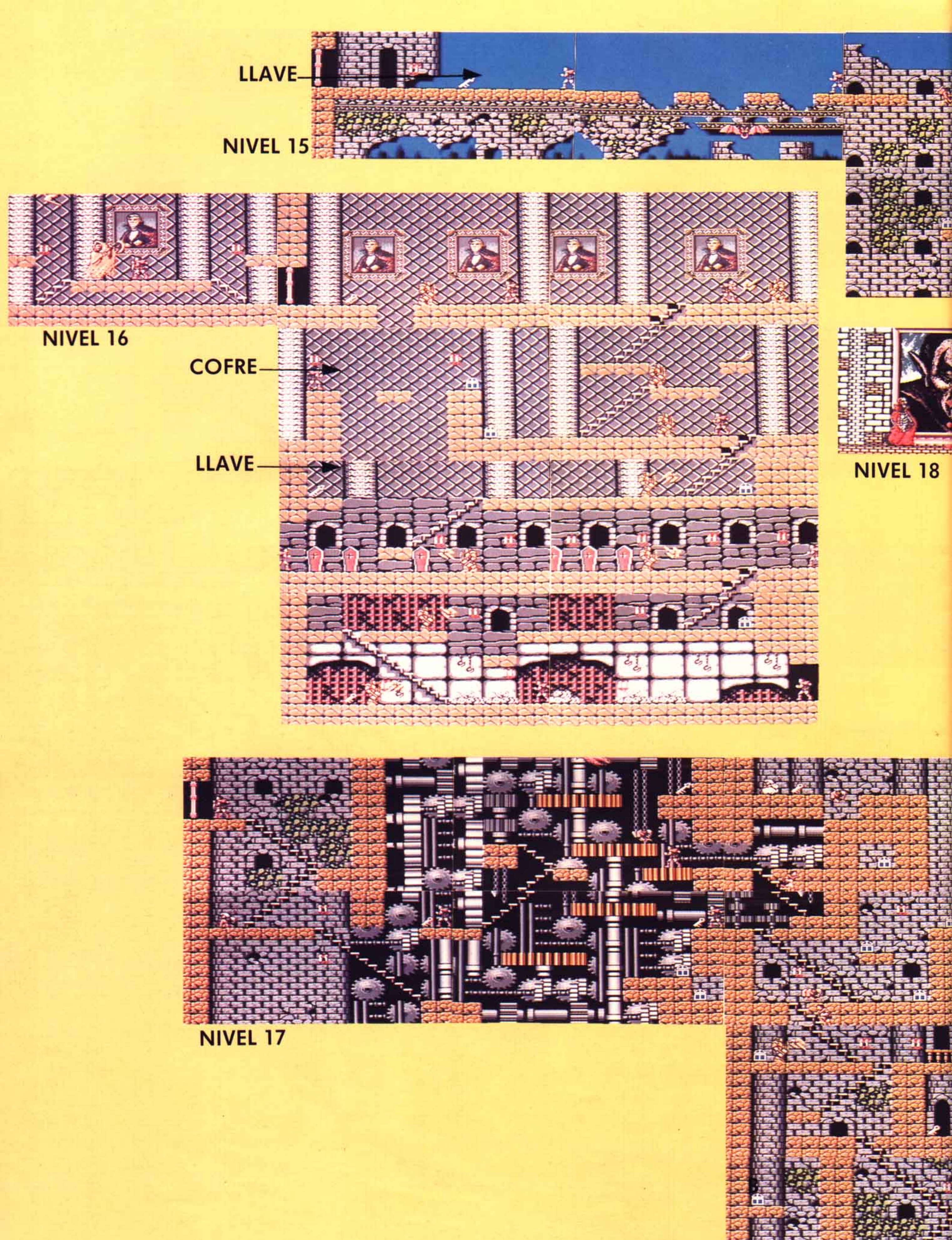
1. Ve hasta el punto A y rompe los dos trozos de falso muro.
2. Coge una llave para abrir cofres.
3. Ve hasta el punto B, y abre el cofre. No pierdas el arma que te acaban de dar.
4. Ve hasta el punto C y destruye el bloque de falso muro que indicamos con la letra D.
5. Recoge la llave y retrocede por donde has venido.

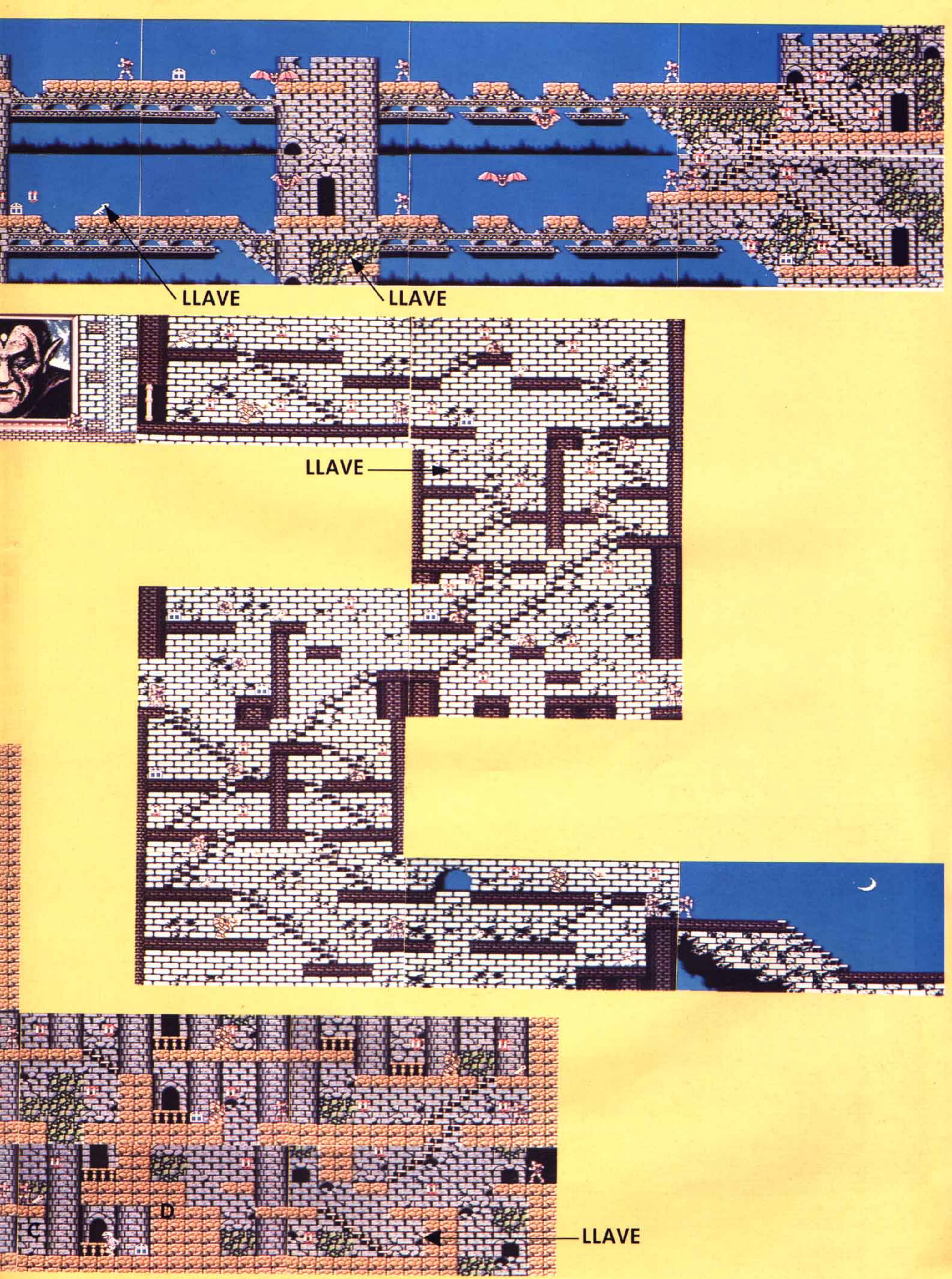
Una vez tengas la llave puedes ir sin ninguna dificultad hasta la puerta y acceder al último nivel.

En el último nivel, el primer acto a realizar es armarte con las espadas. Las conseguirás si tienes suficientes corazones como para comprárselas a la bruja que hay en la segunda pantalla de este nivel. Luego coge la llave y sal del nivel: no hay peligros que no puedas superar por ti mismo. Una vez en la Cámara del Mal, date por muerto si no tienes las espadas, pero si aún las conservas ve de derecha a izquierda para esquivar los ataques del **Conde Drácula**, y a la vez ve disparándole a la cabeza. Una vez acabes con él, sube por el muro y dispara sobre el diamante que hay en la frente del retrato. Si acabas con Drácula habrás llevado la paz a Transilvania.


LLAVE
NIVEL 10
LLAVE
NIVEL 11
B
C
A
A
B
E
LLAVE
NIVEL 12
D
E
D
C
LLAVE
NIVEL 13
NIVEL 14


COFRE
BOLA DE ENERGIA
LLAVE


LLAVE
NIVEL 15
NIVEL 16
COFRE
LLAVE
NIVEL 18
NIVEL 17
B
A



LLAVE
LLAVE
LLAVE
C
D
LLAVE

TODO SOBRE...

PHANTOMAS II

Vuelve de nuevo **Phantomas,** el más famoso ladrón biónico de todos los confines conocidos. Tras lograr su objetivo de saquear la mansión del archimultimillonario y tacaño Goldter, le han encomendado una misión de la cual no sacará un provecho material pero que le granjeará el agradecimiento por parte de la sociedad.

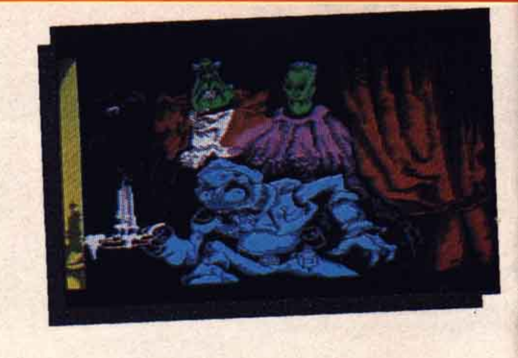
Esta misión consiste en, nada más y nada menos, que eliminar de la faz de la Tierra al elegante y diabólico **Conde Drácula.** Tal noticia le sentó como un tiro a bocajarro en las entrañas.

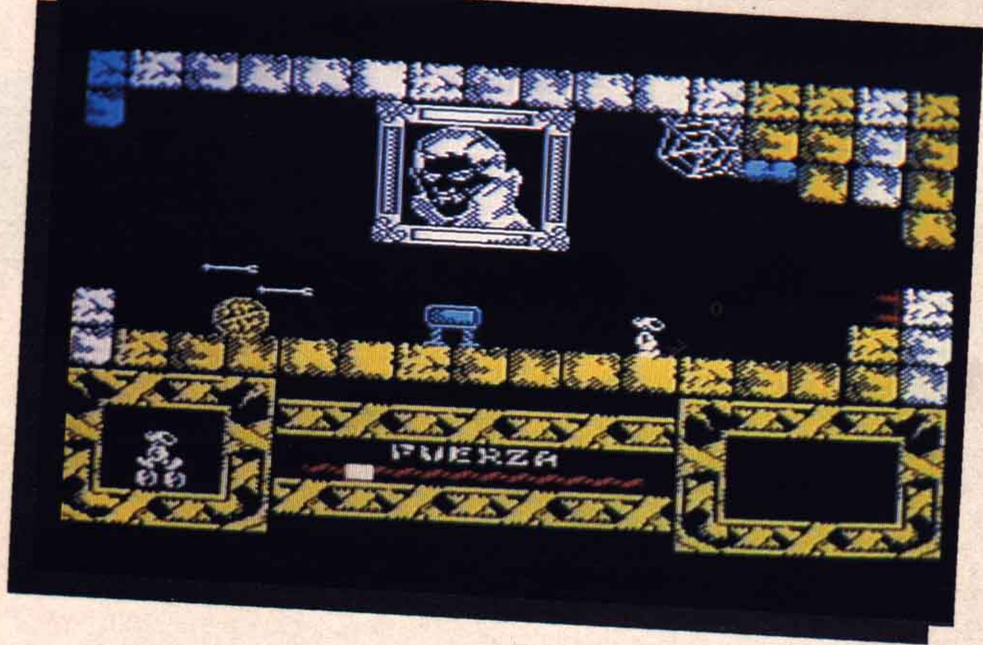
Pero no por ello su frío, exacto y lógico cerebro dejó de funcionar, todo lo contrario. Lo primero que hizo fue dirigirse al banco de datos central para recabar toda la información posible sobre el **Conde Drácula,** y la misión específica que tenía encomendada. El primero y principal problema consistía en que el **Conde Drácula** era un ser muerto, por tanto, no se le podía matar con armas convencionales como HRGP-100 o el haz de rayos gammas positrónicos. Ni con el terrible CLFAO.10N, el gran cañón láser.

Así que **Phantomas** desechó utilizarlos.

Siguió escudriñando información a través del banco de datos. Todos ellos versaban sobre libros antiquísimos que databan del s. XV. O sea, mil años antes de que fuera creado **Phantomas.** Ojeó con suma atención toda aquella superchería y ritos antiguos, pero tuvo que aceptar al final la tradición, puesto que era la única que conocía el método para aniquilar al **Conde Drácula.** Esta consistía en clavar una estaca al **Conde Drácula** cuando éste se hallara durmiendo en su ataúd, mientras, a la vez, se le colocaba sobre su frente la cruz salvadora. Para que el **Conde Drácula** descansara hacía falta que sintiera los rayos del sol, pero su castillo-fortaleza se hallaba sumido en la más absoluta oscuridad. Así que lo primero que debería hacer **Phantomas** era abrir todas las ventanas del castillo para que entrara el reconfortante calor del astro rey.

Con estos datos fijó claro su objetivo. Atacar al **Conde Drácula** en su castillo. Acto seguido pidió información sobre la guarida-fortaleza del noble. Y aunque no era mucho lo que sabía, supo sacar su jugo a la escasa información suministrada por los locos que intentaron llevar a cabo tan peligrosa misión. El castillo, aunque había sido construido hace un milenio, el conde lo había apertrechado de trampas cada vez más modernas y mortíferas. Todo él se hallaba plagado de inmensidad de enemigos, tanto naturales como artificiales, hechos para tal cometido. Y aunque **Phantomas** no tenga vida como el **Conde Drácula,** sí que estos enemigos desgastarán nuestras preciadas reservas de energía concentrada.



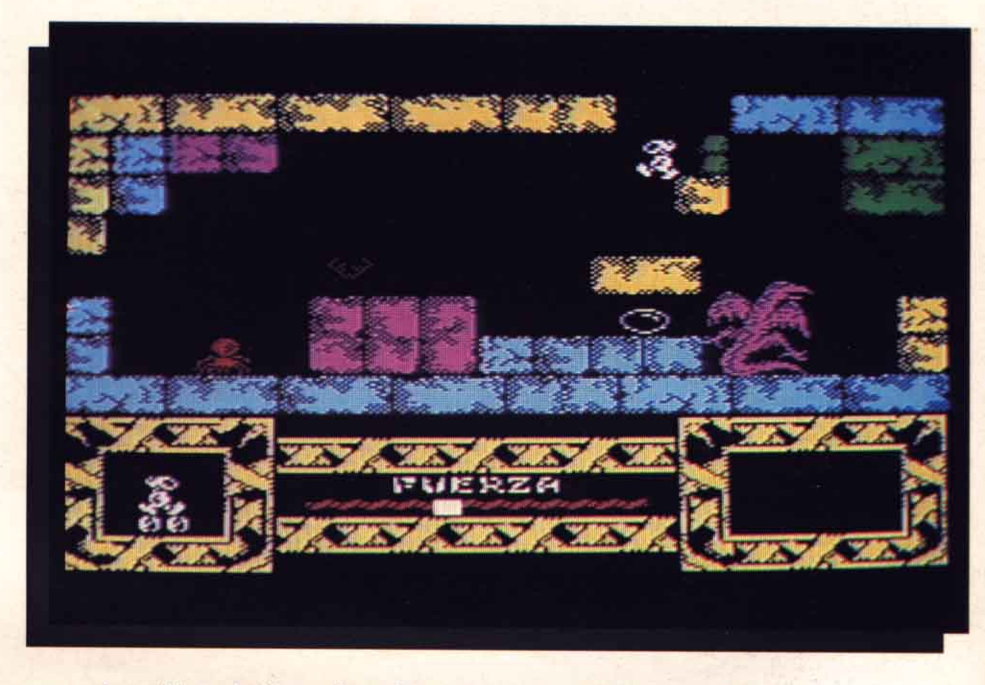


Surtidores de flechas son activados cuando pasamos por delante de ellos. Hay habitaciones donde existen gran número de ellos, por lo que habrá que tener mucho cuidado, pues nuestra energía bajaría a pasos agigantados.

Surtidores de rayos láser. Son la versión moderna de los surtidores de flechas, y se comportan exactamente como sus primitivos modelos. Y aunque **Phantomas** tenga una coraza de energía absorbente LEDAVI, no podrá absorber un rayo de éstos por mucho tiempo, y menos cuando nos pillen en fuego cruzado, pues desembocaría en el gasto de una de nuestras fuentes de energía.

Por todos los objetos fluye la energía negativa que concentra el castillo. Esta, cuando llega a su plenitud, suele salir por los objetos de metal como las copas o los candelabros produciendo un fuerte cortocircuito seguido de unas llamas que si nos sorprenden en nuestros caminos nos harán perder energía, puesto que deberemos activar nuestra pantalla protectora MIANTO-A.

Si todo esto nos parece ya suficiente, aún quedan los peligros mortíferos. Estos peligros, si nos aciertan, acaban con una de nuestras vidas ipso



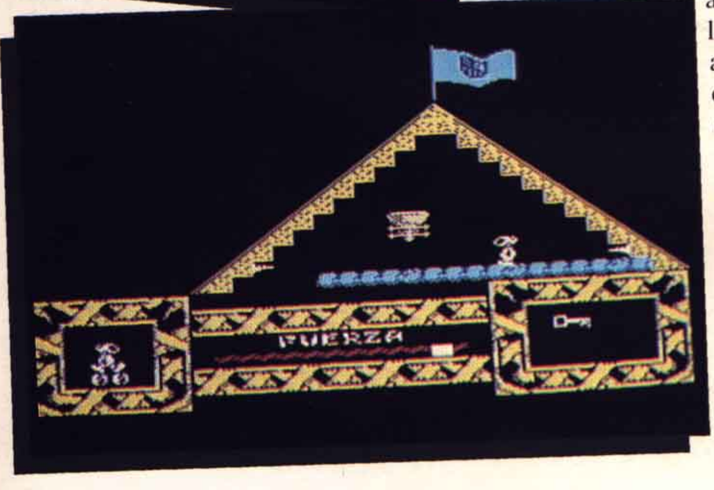
facto, así que deberemos de tener mucho ojo y atención a los movimientos que realizaremos. Estas son piedras que penden del techo y que al pasar sobre ellas se abalanzan y caen sobre nosotros. También hay bolas de energía negativa archiconcentrada, por lo que deberemos tener cuidado de que no nos rocen lo más mínimo. Si éstos son los peligros que existen en el castillo, éste en sí constituye la más peligrosa y mejor defensa para su dueño. Está hecho de forma laberíntica, así que resulta muy fácil perderse en su interior. Hay muros falsos que deberemos abrir y atravesar. Y por último las mortales trampas. Si caemos en una de ellas no podremos salir nunca más y seremos acribillados por todos los demás enemigos.

Pero no todo van a ser inconvenientes. Hay también unas cuantas ayudas de las que podremos sacar partido. Es preciso mencionar las vagonetas, que podemos usar para subir o bajar por las diferentes galerías del castillo. Otra ayuda son los objetos que algún día trajo algún osado y que nos permitirán derribar muros falsos y abrir puertas que de otro modo serían inaccesibles. Y, por último y más importante, el castillo se halla repleto de comida con la cual nos podremos alimentar y reponer energía, ya que la empresa es francamente durísima.

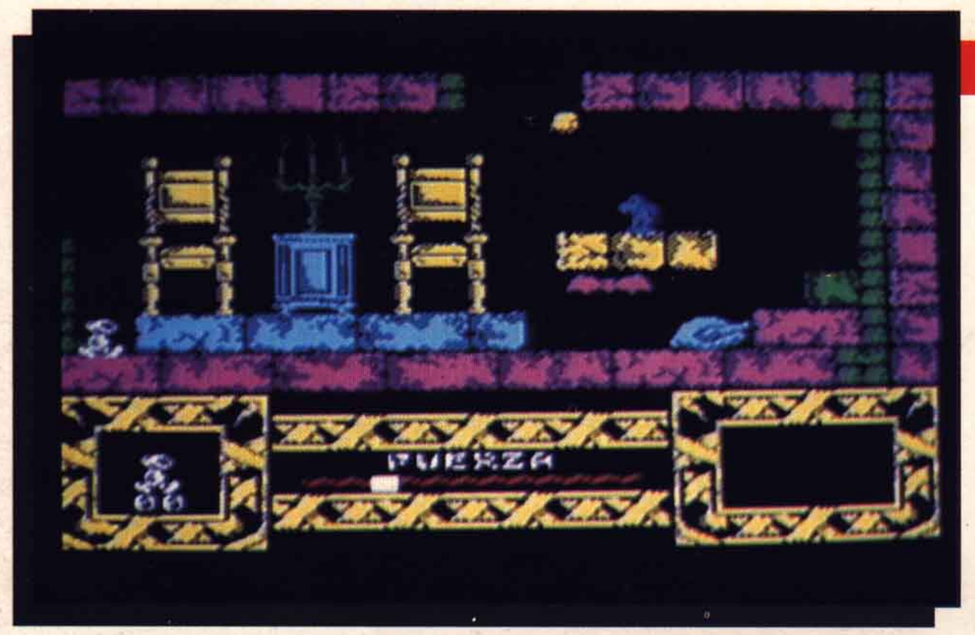
Con todas las ideas claras y ordenadas en su sofisticado cerebro, por fin **Phantomas** se despide de su creador, de los que le han encomendado la labor y de los que han salido a apoyarle moralmente, sube en su vehículo aeroterrestre, y se dirige hacia SILKSTONE, donde comienza su misión: eliminar al **Conde Drácula.**

COMIENZA LA MISION

Al igual que ocurría en la primera parte de **Phantomas,** lo primero que debemos hacer es abrir las ventanas accionando unas palancas. Una vez abiertas, nos dedicaremos a recoger unas llaves, exactamente tres, que se distinguen porque una es triangular, otra redonda y la última, cuadrada. Debemos colocarlas en unos cerrojos que llevan su misma figura, los cuales accionarán el mecanismo que permite levantar muros falsos para seguir la aventura por el interior del castillo. También deberemos recoger una cruz que utilizaremos después. Todo esto lo tenemos que realizar en la parte superior del castillo.



Para acceder a la mansión-fortaleza tenemos que procurar no caer en los fosos que hay en la primera y segunda pantalla. Acto seguido debemos coger una estrella, la cual hará que tocando al contrario se nos abra la puerta del castillo y podamos acceder a él. Una vez en el interior, la primera llave que tenemos que conseguir se halla en la almena inferior derecha del castillo. Es la llave cuadrada. Una vez la llave esté en nuestra posesión nos hemos de dirigir a la segunda pantalla del segundo piso, donde encontraremos el cerrojo de la correspondiente llave y que nos permitirá acceder a 10 pantallas ocultas. Por supuesto, debemos abrir todas las ventanas que vayamos encontrando. Subiremos a la almena más a la izquierda y cogeremos la llave triangular. Con ella nos dirigiremos a la almena 2 de la izquierda y cogeremos la cruz. Ahora nuestros pasos nos llevarán a la pantalla que está justamente debajo de la almena superior derecha y dejaremos su llave en el cerrojo, con lo cual podremos acceder a la almena para recoger el pergamino. Nos iremos a buscar la llave redonda a la habitación más a la derecha del



piso tres. Con todo ello nos dirigiremos a la planta baja. En la habitación número 5 usamos la llave redonda y pasamos a la habitación 6 y de allí al sótano 1; tenemos que recorrer todo el sótano hasta la pantalla 3 contando los fosos. Allí dejamos el pergamino y accedemos a las mazmorras.

Esta es la segunda parte del castillo. Aquí deberemos extremar las precauciones. Tenemos que conseguir la biblia y una tuerca. El mazo y la estaca los dejaremos para la vuelta.

Una vez que estemos en la mazmorra 1, deberemos elegir entre ir por la derecha o hacia abajo. Preferiremos el camino de la derecha, hasta la habitación 5 de la mazmorra 1. Allí bajaremos a la mazmorra 2. Aquí otra vez tenemos que elegir entre ir hacia abajo o continuar por la derecha. Elegimos continuar por la derecha y de ésta hacia abajo —mazmorra 3, habitación 6—, donde encontraremos la biblia. Volvemos hasta M1-H4, tiramos hacia abajo y vamos a la izquierda M2-H3, donde habrá que tener mucho cuidado de no caer, puesto que sería nefasto. Continuamos a la izquierda hasta M2-H2, donde está la tuerca. Con la biblia y la tuerca en nuestro poder tenemos que volver hasta M1-H1, a la derecha y otra vez hacia abajo M5-H2, hacia la derecha y otra vez hacia abajo M6-H3. Una hacia la derecha, una hacia arriba, una hacia la derecha, una hacia arriba y dos a la derecha. Estamos en M4-H7, donde usaremos la biblia para poder acceder a la mina.

LA MINA

Aquí deberemos dejar la tuerca para coger la llave en forma de cruz que nos permitirá pasar a la almena central y entrar en el aposento del **Conde Drácula.**

La mina consiste principalmente en una ascensión hasta los pisos superiores a través de las numerosas vagonetas que se hallan moviendo por las distintas habitaciones. Una vez llegado arriba, cosa que a veces nos parecerá desesperante, puesto que nos caeremos repetidas veces, llegaremos a una pantalla en la cual habrá un ladrillo blanco; éste es signo de que en la próxima pantalla deberemos dejar la tuerca y competir con una máquina que es una «chupóptera» de energía. Para combatirla deberemos movernos frenéticamente a izquierda y derecha, como si de un juego de olimpiadas se tratara. Nuestro nivel de energía comenzará a crecer y llegará un momento en el cual seremos empujados hasta el otro extremo de la pantalla, donde al fin cogeremos la ansiada llave en forma de cruz.

LA VUELTA

Con la llave y el crucifijo volveremos a las mazmorras para coger la maza en M5-H6. Después iremos a por la estaca que se halla situada en M4-H1; una vez todos ellos en nuestro poder nos encaminaremos a la almena central, donde usaremos la llave para poder acceder a la estancia donde se halla el ataúd y, una vez allí, con las tres armas bastará saltar sobre el ataúd y...

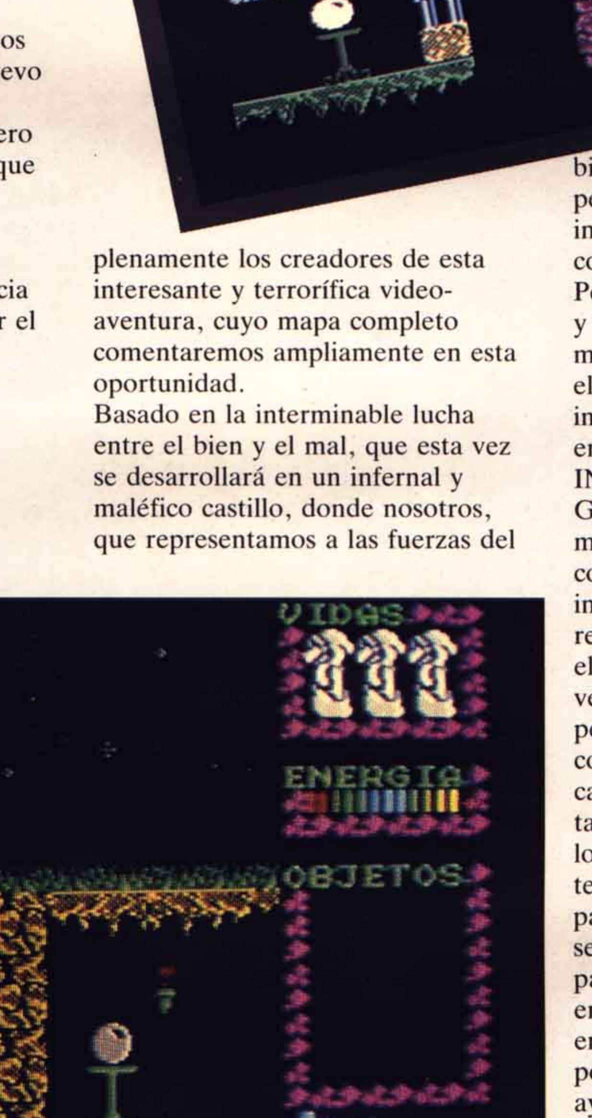
Sí, amigos, aún no ha llegado el final. El alma inmortal del **Conde Drácula** ha dejado de existir, pero no su energía negativa que le acompañó toda su vida. Y está electrizada por la sed de venganza del Conde. Se echa hacia nosotros, hasta que nos consume pero...

Como en toda buena película, el héroe, en nuestro caso **Phantomas**, saca de su bolsillo la terrible HRPG-100 acabando con todo resto de partículas electromagnéticas negatrónicas.

EL CARGADOR

```
10 '
20 ' cargador de PHANTOMAS 2 para MSX
30 ' cargador para INPUT por
40 ' JOSE VILA
50 '
60 COLOR 15,1,1:KEY OFF:CLS
70 INPUT"energia infinita (s/n) ";EI$
80 IF EI$="s" THEN LET EI=1
90 IF EI$<>"s"THEN GOSUB 200
100 INPUT"Inmunidad total (s/n) ";IT$
110 IF IT$="s"THEN LET IT=1
120 IF IT$<>"s" THEN GOSUB 200
130 BLOAD"CAS: ",R
140 BLOAD"CAS: ",R
150 BLOAD"CAS:"
160 IF EI=1 THENPOKE &HA500,201
170 IF IT=1 THEN POKE &HA53B,201 :POKE &HA578,201 :POKE &HA56B,201
180 DEFUSR=&H8598
190 A=USR(0)
200 LET EI=0:LET IT=0
210 RETURN
```

MAPA DE SPIRITS



Si el mes pasado os hacíamos entrega del cargador del programa SPIRITS de TOPOSOFT, este mes os ofrecemos su sensacional mapa con las más de 80 pantallas que lo forman.

En la carrera por los programas de laberintos y arcades iniciados en el MSX por el programa SORCERY, ahora TOPOSOFT, recogiendo el guante, ha sabido crear un programa donde han sabido incorporar las mejores ideas de los programas anteriormente hechos, pero sin olvidar de poner nuevos elementos que hicieran de éste un juego nuevo y distinto.

Como comentábamos en el número anterior de INPUT MSX, en el que otorgamos a esta video-aventura nada menos que una calificación total de 42 puntos, lo que en realidad aporta interés y relevancia a la apasionante tarea de innovar el software son los cambios que afectan directamente al planteamiento de la acción del programa, o al desarrollo en la pantalla de la idea original del guionista.

Y este objetivo lo han logrado plenamente los creadores de esta interesante y terrorífica video-aventura, cuyo mapa completo comentaremos ampliamente en esta oportunidad.

Basado en la interminable lucha entre el bien y el mal, que esta vez se desarrollará en un infernal y maléfico castillo, donde nosotros, que representamos a las fuerzas del bien, deberemos proteger a dos personajes que desde nuestra infancia nos fueron fieles compañeros de múltiples aventuras. Pero no lloréis ante tantos recuerdos, y luchad hasta que las fuerzas del mal os obliguen a doblegaros ante ellas (cosa que será muy difícil, casi imposible, si usáis el cargador que se encuentra en el número 13 de INPUT MSX).

Gracias al mapa podrás hacer un mejor uso de elementos mágicos como es la bola de cristal, que te indica dónde se encuentran los restantes personajes del juego. Con el mapa podrás averiguar verdaderamente donde se halla el personaje que estés interesado en controlar, y encontrar el mejor camino para ir hasta él. El mapa también te será de gran ayuda para localizar las diferentes palancas que te sirven para abrir puertas y pasadizos secretos. Pero no todo te será tan fácil como ahora te pueda parecer: en el mapa no aparecen los enemigos a quienes tendrás que enfrentarte, pero recuerda que posees un rayo paralizador que te ayudará a librarte de sus ataques. En caso de ser atacado deshazte rápidamente de tus enemigos, ya que

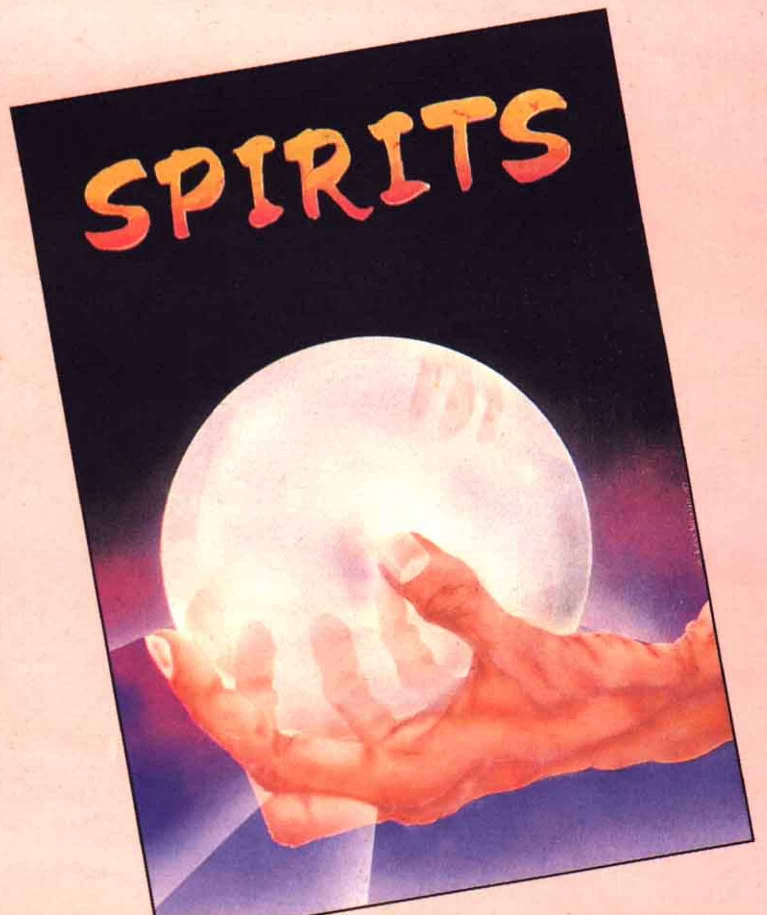


ellos te irán restando energía, y cuando ésta llegue a cero perderás una vida, aunque si has usado el cargador…

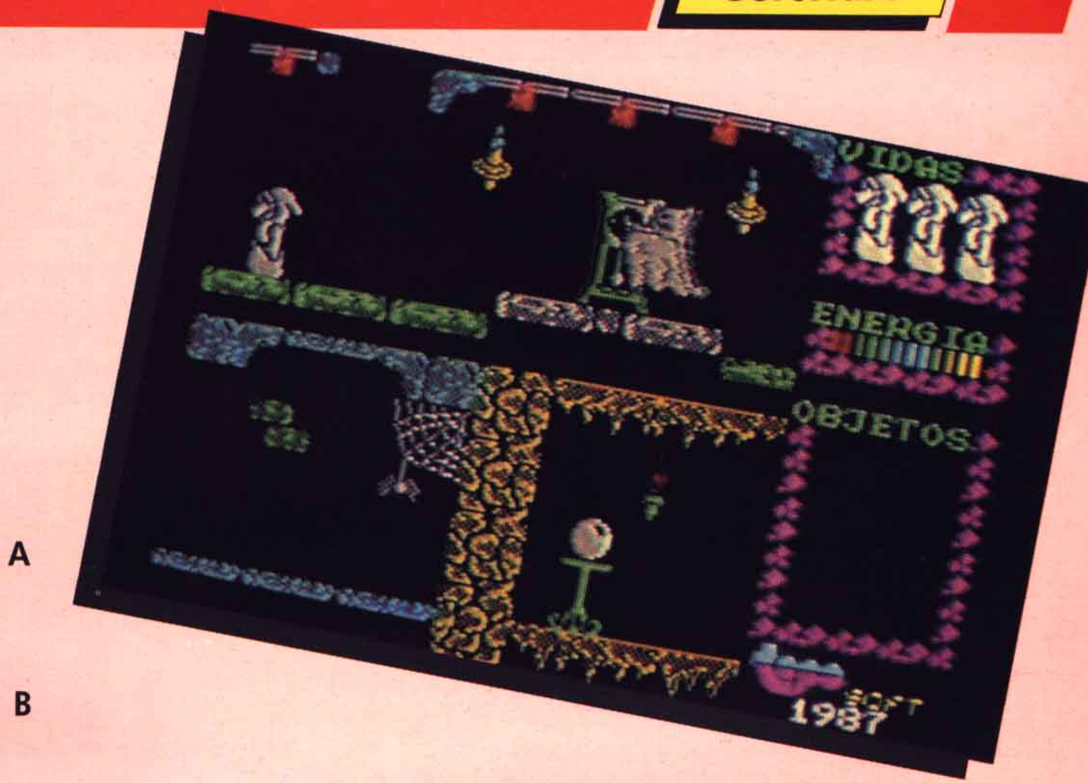
La aparición simultánea de cantidad de información en pantalla quizá te despiste un poco, pero en todo momento recuerda que lo más importante es mantener a salvo al mago que tú conduces por el castillo. En el mapa podrás observar la disposición de todas y cada una de las 87 pantallas que lo forman, y gozar de los espectaculares gráficos que adornan este juego. En el castillo encontrarás de todo: escudos, antiguos pergaminos, calderos, etc., pero recuerda que los tres elementos más importantes, y que tú debes conseguir, son la bola de cristal, antes citada, la varita mágica, imprescindible para desencantar a nuestra bella doncella, y el libro de conjuros, que nos servirá para romper el hechizo que liga a la armadura.

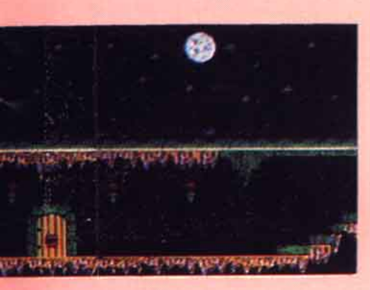
A

B



El mapa de la aventura de este mes se llama SPIRITS. Aquí os lo presentamos fotográficamente para que podáis recorrer sus tétricas mazmorras, que rezuman maldad, olvido y terror, con la tranquilidad de llevar una buena guía bajo el brazo: INPUT. No os perdáis por el camino y pensad que el pasadizo A sigue en A y el B continúa en B. Cosa lógica, ¿no?





A

B

SOFTACTUALIDAD

MACADAM BUMPER

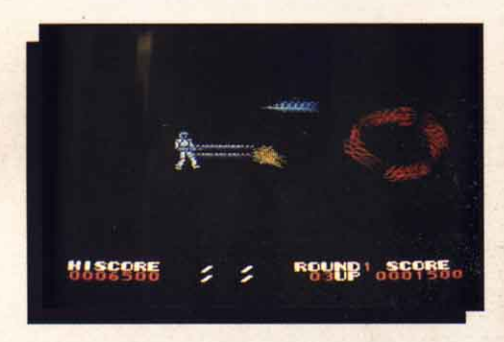
Los PINBALLS fueron las primeras máquinas de las que se pudo disfrutar, y desde que aparecieron en bares y salones recreativos nunca han dejado de evolucionar. Uno de los pasos más importantes de esta ininterrumpida evolución fue la incorporación de equipos informáticos en su confección. Y MACADAM BUMBER es un Pinball ciento por ciento informatizado.
Al cargar el programa nos aparece en pantalla una atractiva máquina donde podemos jugar increíbles partidas. Los efectos son muy reales y hasta podemos simular que golpeamos, pero deberemos ir con cuidado de no provocar el terrorífico TILT. Las dos cualidades más importantes de MACADAM BUMPER, y que convierten a éste en un programa y no en un mero juego, son la posibilidad de variar los parámetros y la de la confección de nuestras propias máquinas. Variar los parámetros nos permite cambiar la velocidad de la bola, la fuerza de rebote, la cantidad de puntos a conseguir para que se nos recompense con una bola extra, etc. La posibilidad de crear una máquina a nuestro gusto es la más completa y espectacular. Al acceder a esta opción veremos en la parte izquierda de la pantalla todos los elementos de que disponemos para formar la máquina, y cada uno de ellos estará representado por una letra. En la derecha de la pantalla hay todo un espacio libre y un cursor: para confeccionar nuestra máquina tan sólo hace falta mover el cursor hasta donde queramos colocar cada pieza y, una vez allí, apretar la letra que representa dicha pieza, y ésta aparecerá donde teníamos el cursor. Una vez tengamos montada toda la máquina, podemos colorearla, también a nuestro gusto. Cuando todo esté acabado, podemos grabar la máquina que hemos confeccionado en una cinta y así recuperarla para posteriores partidas.
Ésta es una atractiva oferta, pues el trabajo realizado queda recompensado al jugar en nuestra propia máquina. Vuestra imaginación también participa de este programa. Y vuestro bolsillo lo agradecerá todos los domingos por la tarde.





ALFA-ROID

ALFA-ROID es un arcade que tiene como protagonista a un apuesto guerrero intergaláctico. Éste debe ir por el espacio acabando con todos los enemigos que se interpongan en su camino. Esta misión la realiza sin nave alguna, ya que nuestro héroe es capaz de volar, y de forma parecida



al norteamericano que salió por los aires en la inauguración de los Juegos Olímpicos de 1984 en Los Angeles. En su viaje espacial, nuestro guerrero puede entrar en cavernas ocultas donde deberá enfrentarse, en un combate de artes marciales, a otros guerreros. Estos combates son a vida o muerte, y la energía de cada uno de los dos oponentes viene representada por un gráfico en la parte inferior de la pantalla. Dicho gráfico tiene segmentos coloreados de verde, azul, amarillo y rojo. Cada vez que uno de los dos contendientes logra golpear al otro, le resta energía, y ganará aquel que acabe antes con toda la energía del enemigo. Si pierde ALFA-ROID, todo habrá terminado, pero si gana, además de poder continuar viajando por el espacio, se le hará entrega de un arma adicional, que aumentará su poder destructivo.
Éste es un programa que ha querido unir dos juegos de gran éxito, los arcades espaciales y los basados en las artes marciales, como el judo, kárate o la lucha libre. Y la verdad es que el resultado ha sido mejor de lo que se podía esperar, ya que unir dos conceptos tan diferentes de formas de luchar es muy difícil. El éxito reside en que se ha sabido

separar claramente estas dos facetas del juego, pero a la vez se las ha unido de manera que los resultados obtenidos en una de ellas repercutieran sobre la otra. Por tanto, no se trata de dos juegos por separado, sino de la perfecta simbiosis de ellos en un solo programa.



SKOOTER

Son muchos los juegos que tienen como único objetivo poner a prueba nuestra habilidad y reflejos para superar a unos cuantos bicharracos que nos quieren matar. Pero hay algunos que, además, nos hacen pensar, y SNOOKER es un programa que ha sabido unir estas dos características, arcade y lógica, perfectamente. Un aspecto parece complementar al otro y no estar en contra, como pasa a menudo en muchos juegos. El objetivo del programa es lograr coger una serie de piezas que se encuentran en la pantalla, o nivel, donde estamos, sin ser alcanzados por nuestros enemigos. Para evitarlos debemos trazar una estrategia que nos permita, a la vez que salir ilesos, haber recogido las bombillas, tuercas, velas, etc. que había por la pantalla. Por desgracia, podemos convertirnos en nuestros peores enemigos, y para los casos en que quedemos atrapados por no haber pensado demasiado, podemos autodestruirnos usando la tecla F5. Otro aspecto muy destacable del programa es la gracia con que está hecho. Por un lado los gráficos son casi perfectos, vistosos y con un alegre colorido. La música que nos acompaña en todo momento es animada y si, por desgracia, nos matan, un divertido efecto sonoro hará que no nos enfademos. Y por otro lado entre nivel y nivel se nos muestra una cita literaria, eso sí en inglés, que de algún modo es una pista. SKOOTER gustará de seguro a aquellos que les gustó EGGERLAND MISTERY, pero encontrarán a faltar los códigos de acceso directo. En cambio otros elementos, como las flechas que sólo permiten el paso en una dirección o los bloques de piedra móviles, son comunes a los dos juegos. BYTEBUSTERS ha vuelto a demostrar que no quieren especializarse en un tipo concreto de juegos, y que su oferta es amplia y variada.

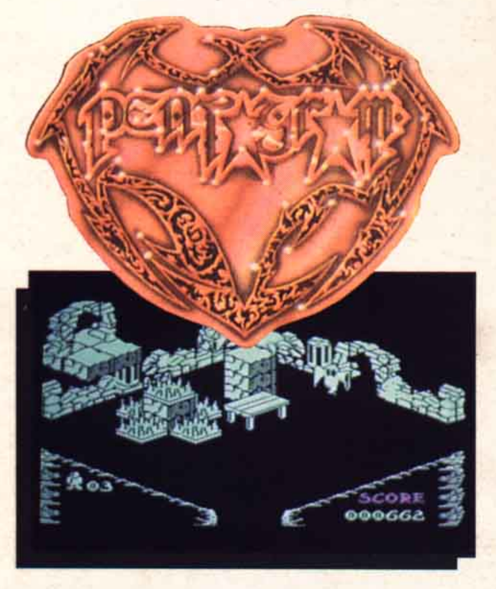
PANEL PANIC

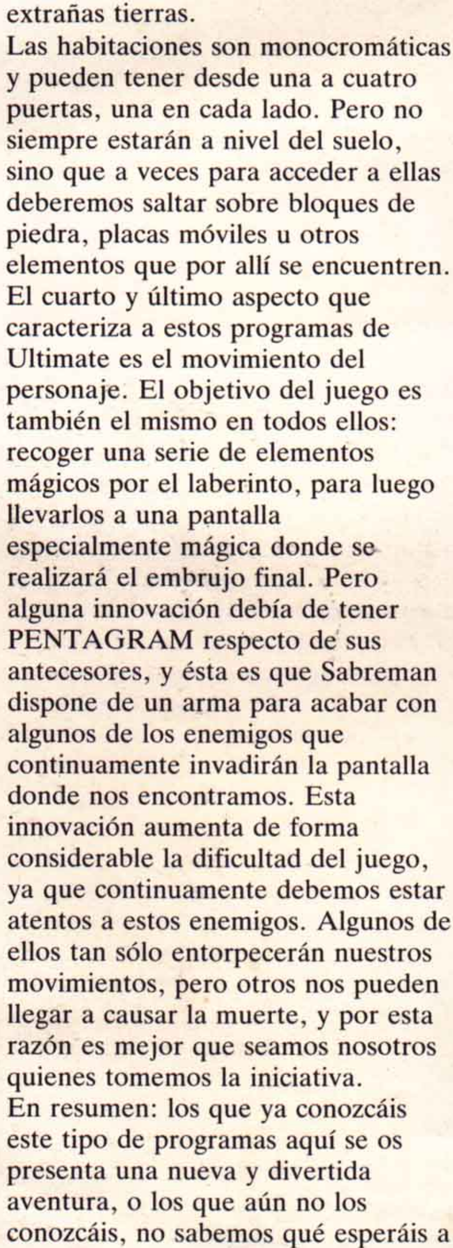
Dentro de la serie de programas que la Bytebusters ha sacado bajo el calificativo de CLASSIC 2, este mes os destacamos PANEL PANIC, un entretenido juego que es seguro que todos ya conocéis. El objetivo del programa es colorear toda la pantalla. La cuestión es que ésta se halla dividida en rectángulos que, al pasar por los cuatro lados que los forman, se llenan de color. Tan sólo podemos desplazarnos por las líneas que determinan a los rectángulos. Y por ellos también circulan nuestros enemigos, a los que, lógicamente, debemos esquivar. Además, para desgracia nuestra, no hay forma de matarlos y siempre los tendremos detrás de nosotros.
Éste es un programa que no destaca por su calidad gráfica, sino por el atractivo del juego en que se basa, un juego que tuvo un gran éxito en su versión para máquinas recreativas. El juego es simple pero su adicción es poderosa: nos hará pasar delante de la pantalla, horas interminables.

PENTAGRAM

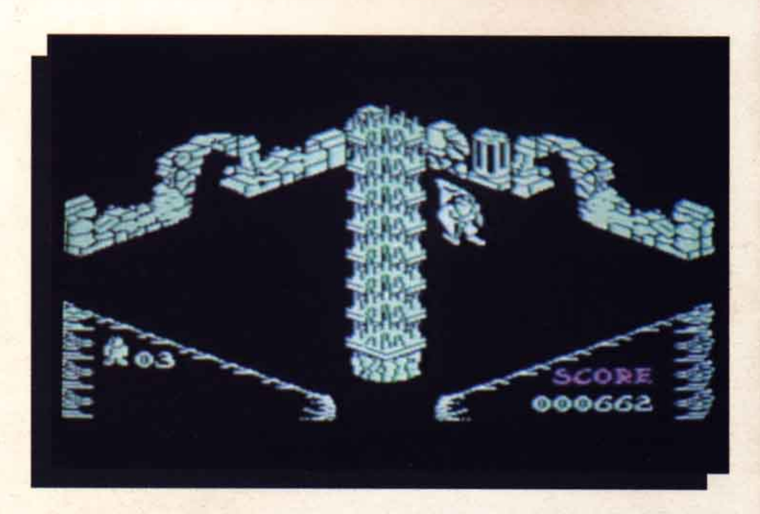
● ULTIMATE ■ JUEGO ▲ CASSETTE

Podemos decir, sin miedo a equivocarnos, que son muchos los usuarios del MSX que anteriormente han tenido un Spectrum. Y muchos de éstos aún deben creer que los mejores programas para ese ordenador son los que hizo ULTIMATE. Y fue este fabricante de software quien revolucionó el mercado con el método *filmation*. Y a partir del éxito de su primer programa, hecho según esta técnica, KNIGHT LORE, ha sacado diferentes versiones del mismo, como lo fue en su época ALIEN 8. Y ahora lo es PENTAGRAM. Ésta es la nueva aventura en que está metido Sabreman, protagonista de todas las aventuras de Ultimate. Su nueva misión es encontrar a los Runo, y recopilarlos, ya que entonces éstos encontrarán su lugar en la magia y llevarán a Sabreman hasta el codiciado PENTAGRAM.
Pero para los que aún no conozcáis este tipo de programas os diremos que sus principales características son cuatro. En primer lugar, la representación tridimensional de los escenarios. La segunda característica es la posibilidad de mover algunos de los objetos que hay en las pantallas. El tercer aspecto de gran importancia es que toda la acción se desarrolla dentro de un laberinto. Y este laberinto está dividido en diferentes habitaciones por las cuales debemos ir buscando los Runo, y donde vamos a tropezar con los maléficos habitantes de tan extrañas tierras.





Las habitaciones son monocromáticas y pueden tener desde una a cuatro puertas, una en cada lado. Pero no siempre estarán a nivel del suelo, sino que a veces para acceder a ellas deberemos saltar sobre bloques de piedra, placas móviles u otros elementos que por allí se encuentren. El cuarto y último aspecto que caracteriza a estos programas de Ultimate es el movimiento del personaje. El objetivo del juego es también el mismo en todos ellos: recoger una serie de elementos mágicos por el laberinto, para luego llevarlos a una pantalla especialmente mágica donde se realizará el embrujo final. Pero alguna innovación debía de tener PENTAGRAM respecto de sus antecesores, y ésta es que Sabreman dispone de un arma para acabar con algunos de los enemigos que continuamente invadirán la pantalla donde nos encontramos. Esta innovación aumenta de forma considerable la dificultad del juego, ya que continuamente debemos estar atentos a estos enemigos. Algunos de ellos tan sólo entorpecerán nuestros movimientos, pero otros nos pueden llegar a causar la muerte, y por esta razón es mejor que seamos nosotros quienes tomemos la iniciativa.
En resumen: los que ya conozcáis este tipo de programas aquí se os presenta una nueva y divertida aventura, o los que aún no los conozcáis, no sabemos qué esperáis a hacerlo.




```
1Ø  ' cargador del PENTAGRAM
2Ø  ' para INPUT MSX
3Ø  ' por JOSE VILA
6Ø  BLOAD"cas:"
7Ø  KEYOFF:COLOR 15,1,1:CLS
8Ø  POKE &HEØDD,&HF1:POKE
    &HEØDE,&HDE
9Ø  INPUT"numero de vidas (s/n)
    ";A$:IF A$="n"THEN GOTO
    11Ø
95  GOSUB 13Ø
1ØØ INPUT"cuantas vidas
    iniciales (Ø/255) ";B:POKE
    &HDEF2,B
11Ø INPUT "vidas infinitas (s/n)
    ";C$:IF C$="s"THEN
    GOSUB 13Ø:IF
    C$="n"THEN POKE
    &HDEF9,Ø AND POKE
    &HDEFA,Ø
12Ø DEFUSR=&HEØØØ: A=USR
    (Ø)
13Ø FOR N=&HDEF1 TO
    &HDEFD:READ
    A$:A$="&h"+A$:POKE N,
    VAL (A$):NEXT N
135 RETURN
14Ø DATA 3e,Ø5,32,d2,ae,3e,
    ØØ,32,aØ,c5,c3,ØØ,5d
```

ANIMACION	***6***
INTERES	***7***
GRAFICOS	***7***
COLOR	***5***
SONIDO	***5***
TOTAL	***30***



AUF WIEDERSEHEN, MONTY

● GREMLIN GRAPHICS ■ JUEGO ▲ CASSETTE

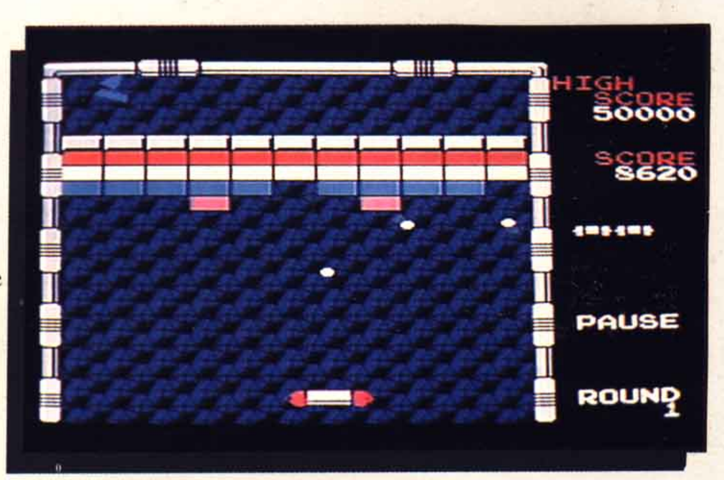
Monty está en el Peñón, y todos los servicios secretos de los países europeos van tras su pista. Él ahora luchará por su libertad e intentará viajar por toda Europa para recolectar suficiente dinero y comprarse la isla griega de Montos. En Montos, Monty confía escapar de la extradición y poder disfrutar por fin de una vida tranquila. Ésta es la historia en que los programadores de Gremlin Graphics han basado esta nueva aventura de Monty, y, según todo indica, la última, pero nunca se sabe. Y toda esta historia queda traducida en un divertido juego en que debes llevar a un simpático topo pantalla tras pantalla, superando todas las dificultades que en ellas se le presenten. Pero con el atractivo de que todo el programa está ambientado en Europa y, por ejemplo, te puedes encontrar intentando vender la *Mona Lisa*, o corriendo en el Grand Prix de Mónaco. Es un programa clásico realizado bajo el más puro estilo de *Jet Set Willy II* o *Dynamite Man*, pero con un toque de humor británico. Humor que no siempre os hará reír. Siempre que Monty pasa de estado suena el himno del estado en que acaba de entrar, y al entrar en el estado español no suena el himno de nuestro estado, ni de ninguna de las naciones que lo componen, sino que se oye el *Viva España* de Manolo Escobar, eso sí, muy bien interpretado. Pero si dejamos de lado estos detalles, casi anecdóticos, podemos disfrutar de un juego que ha sabido recoger la experiencia adquirida en la programación de esta clase de arcades.
A este juego tanto podréis jugar con las teclas como con el joystick.

EL CARGADOR

```
1Ø ' cargador para INPUT MSX
2Ø ' del AUF WIEDERSEHEN
     MONTY
3Ø ' por JOSE VILA
4Ø '
5Ø BLOAD"cas:"
6Ø POKE &HD829,
     Ø:POKE &HD82A,
     &HD9
7Ø FOR N=&HD9ØØ TO
     &HD9Ø8:READ A:
     POKE N,A:
     NEXT N
8Ø DEFUSR=&HD8ØØ:
     A=USR(Ø)
9Ø DATA 229,62,4Ø,
     5Ø,115,169,225,
     233,2Ø1
```

ANIMACION	7
INTERES	7
GRAFICOS	8
COLOR	7
SONIDO	9
TOTAL	**38**





Seguro que habéis jugado una infinidad de partidas con este programa y siempre pensando lo fenomenalmente bien que os iría un cargador que os permitiera jugar con infinitas vidas. Pues aquí lo tenéis, ya es vuestro. Ahora id rápidamente hasta el ordenador, tecleadlo y cargad vuestro arcade preferido: ARKANOID. Pero no penséis que ya está todo hecho, ahora tan sólo os hemos facilitado las cosas para que logréis llegar hasta el final, pero todavía hay muchas dificultades a superar, entre ellas 32 increíbles pantallas. Las otras seguramente ya las conocéis, pero como siempre hay alguno que le cuesta reconocer los buenos programas le vamos a explicar un poco de qué va ARKANOID. Tu controlas Vaus, una nave intergaláctica con la que deberás lograr superar los 32 niveles antes de poder luchar contra el «CAMBIADOR DE DIMENSIONES», a quien deberás vencer en una terrible batalla. En caso de éxito la resurrección de ARKANOID se habrá hecho realidad. Cada uno de los 32 niveles a superar está formado por una pantalla llena de ladrillos, y entre éstos configuran diferentes formas geométricas. Tú con tu nave y gracias a una pequeña bola deberás ir destruyendo los ladrillos hasta que no quede ni uno, o tan sólo esos que son indestructibles (de color amarillo).

La dificultad del juego radica en que si la pelota cae más abajo de donde tú te encuentras te destruyes. Y no será fácil mantenerla en la superficie, ya que poco a poco se va acelerando hasta ir a velocidades mayores que la de la luz, pero seguro que antes ya te han matado.

Pero no todo está en tu contra, a veces al destruir un ladrillo cae de él una cápsula. Cada cápsula es de un color diferente y si la recoges con tu nave te proporcionará diferentes poderes, según el color de ésta. Las más importantes cápsulas son las tres siguientes:

Amarilla: frena la velocidad de la pelota, facilitando su movimiento.

Roja: arma a nuestra nave con un potente láser que nos permite destruir los ladrillos más rápidamente.

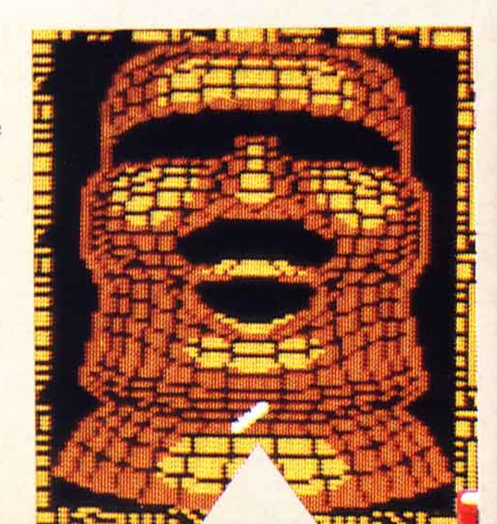
Violeta: rompe una sección de pared por donde podemos acceder directamente al siguiente nivel.

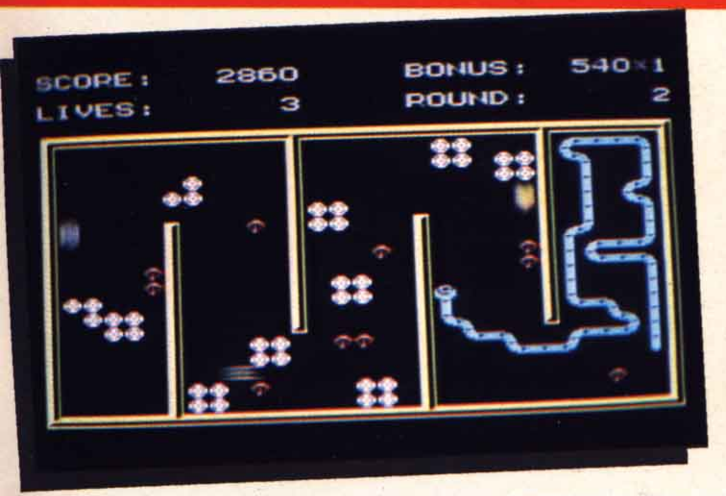
La cápsula gris nos recompensa con una nave extra, pero como ahora tenemos infinitas naves podemos prescindir de ella. Y la cápsula azul claro será en algunos niveles la más importante de conseguir, ya que ella hace aparecer en pantalla dos bolas más.

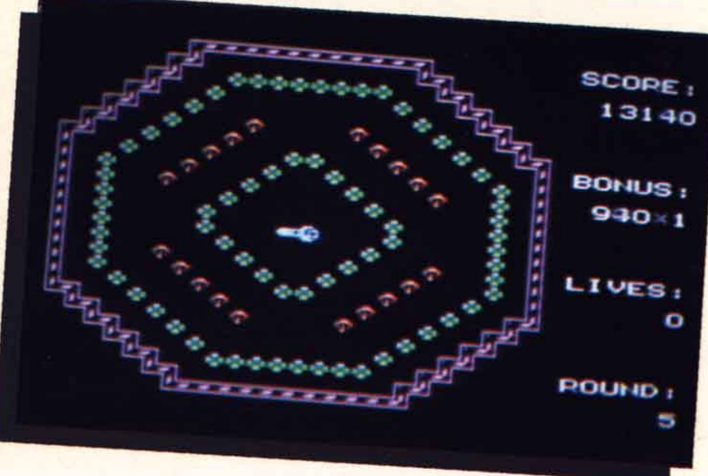
Para acabar recordaremos los alienígenas que aparecen por la parte superior de la pantalla. Éstos tanto pueden facilitarnos como dificultarnos las cosas. Los alienígenas se destruyen cuando nuestra bola impacta con ellos, entonces ésta cambia de dirección y esto puede causar catástrofes, o nos puede ayudar a completar un nivel. Bueno esperamos que esta vez ya logréis llegar hasta el «CAMBIADOR DE DIMENSIONES» y le ajustéis las cuentas por haberos tenido en vela tantas noches.

EL CARGADOR

```
1Ø ' cargador de ARKANOID
2Ø ' para INPUT MSX
3Ø ' por JOSE VILA
4Ø '
5Ø BLOAD"cas:"
6Ø COLOR 15,1,1: CLS:KEYOFF
7Ø INPUT"Vidas infinitas (s/n)";
   A$
8Ø IF A$="s" THEN GOSUB 1ØØ
9Ø DEFUSR=&HCØØØ:
   A=USR(Ø)
1ØØ POKE &HCØ3F, &H58:POKE
    &HCØ4Ø,&HCØ
11Ø FOR N=4924Ø! TO 49246!:
    READ A:POKE N,A:NEXT N
12Ø RETURN
13Ø DATA 62,Ø,5Ø,16,124,195,
    42,64
```





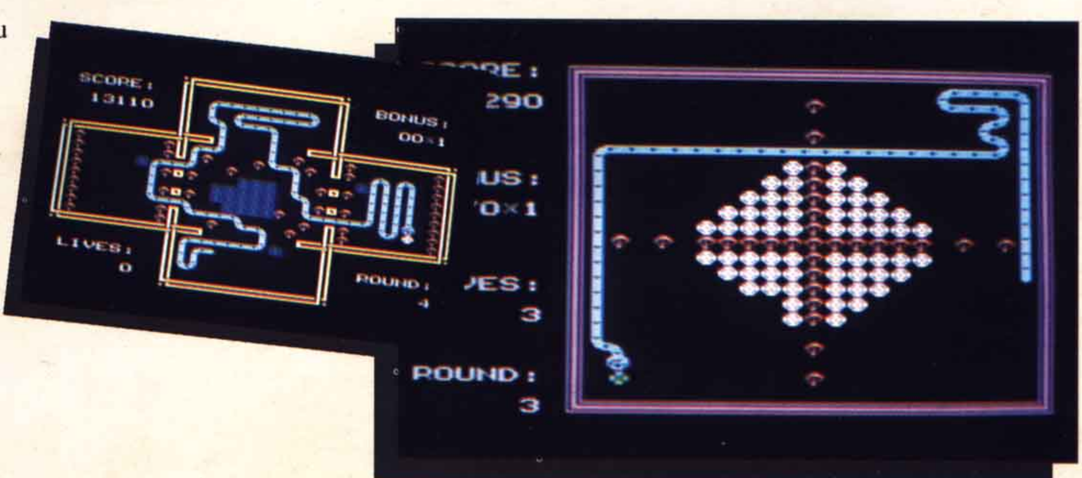


SNAKE IT

● AACKOSOFT ■ JUEGO ▲ CASSETTE

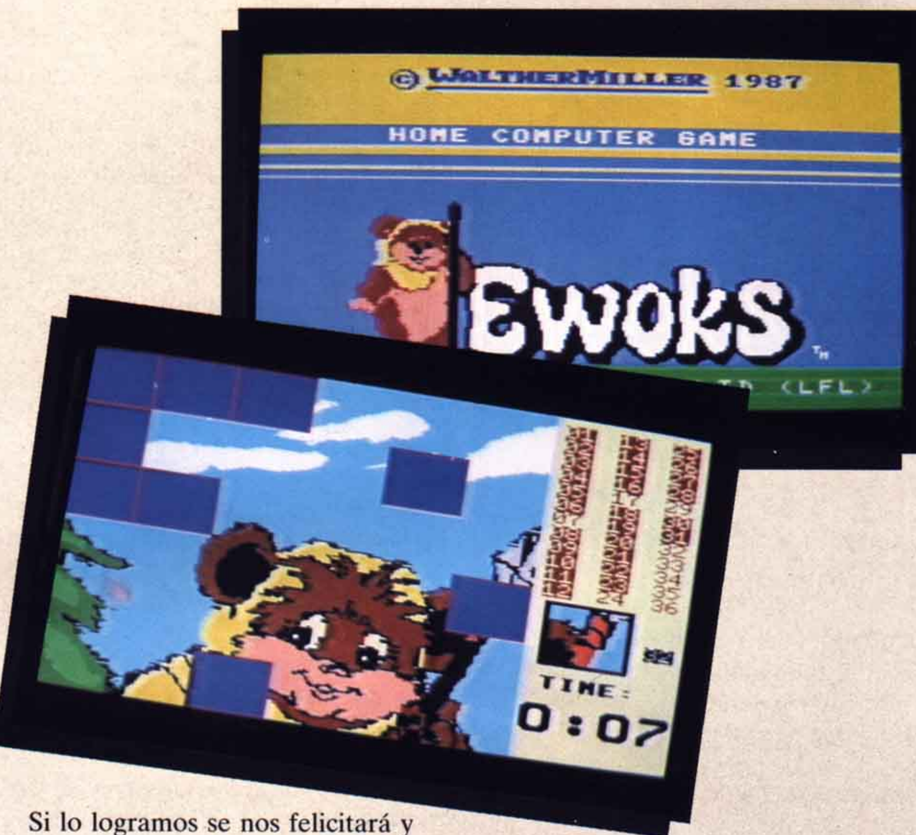
Éste es un juego diferente a la mayoría de los que podamos encontrar en el mercado. No va de matamarcianos, de simuladores de vuelo, ni de ajedrez o algún deporte más o menos conocido, sino que se basa en una pequeña serpiente que debe ir comiendo las diferentes plantas que hay en la pantalla. Pero no todas las plantas: el juego tiene sus reglas. Las plantas verdes y blancas son las que debemos comer, y a las rojas mejor ni acercarse, son mortales. Pero no todo acaba aquí; al principio, sólo veremos plantas verdes y rojas en la pantalla, y las blancas aparecerán cuando empecemos a comernos a las verdes. Y no sólo esto, si no que luego cada vez que nos comamos una planta blanca aparecerá una roja, y aún no acaba aquí todo, ya que al comer las plantas verdes nuestra pequeña serpiente habrá crecido tanto que ahora deberá ir con cuidado de no entorpecerse en su propio camino. Si queda atrapada por su propio cuerpo la única solución será que la matemos pulsando ESC, cosa que da mucha rabia de hacer.
Lo qùe parecía un sencillo juego para niños se acaba de convertir en un emocionante juego donde las prisas y los nervios acabarán por traicionarnos. Y es que el juego va aumentando en emoción y dificultad a medida que pasamos de pantalla, cada vez todo está más y mejor pensado, de manera que es imposible ganar sin una estrategia preestablecida. Y, por si esto no fuera poco, de tanto en tanto aparecen por la pantalla diferentes personajes que nos dan una puntuación extra, nos duplican o triplican los bonus. Esto ya es una tortura porque por un lado queremos coger a estos corazones o bolas que harán aumentar nuestra puntuación, y por el otro lado sabemos lo peligroso que es ir tras ellos (¿no os habríais creído que se estaban quietos?), ya que podemos acabar muertos tras comer una seta roja. En definitiva, no nos van a dejar pensar en paz. Todo está en contra de nosotros, y si además algún listorro de esos que no se saben callar la boca nos está observando, acabaremos teniendo que ir al psiquiatra o pegándole una... Es que tan sólo de pensar en este juego ya nos ponemos nerviosos. O sea que mejor cambiar de tema. Los gráficos son sencillos, pero tienen mucho atractivo, sobre todo por los colores escogidos y lo que representa cada uno de ellos. Otro aspecto que vale la pena destacar de este programa es que al acabar de ser grabado nos saluda con una frase sintetizada, y más tarde podremos oír un par más de frases, una de ellas la odiosa: GAME OVER.
Os advertimos de que si compráis este programa para vuestros hijos seréis vosotros quienes acabaréis delante del ordenador. Y es que no os podréis estar tranquilos sin ayudarles un poquito, y luego lentamente, como quien no lo quiere...
Un juego rápido, adictivo, de una lógica aplastante que hará las delicias de los mil y un aficionados a los rompecabezas mentales.

ANIMACION	7
INTERES	8
GRAFICOS	6
COLOR	6
SONIDO	7
TOTAL	**34**



EWOKS Y DROIDS

● WALTHER MILLER ■ JUEGO ▲ CARTUCHO ROM

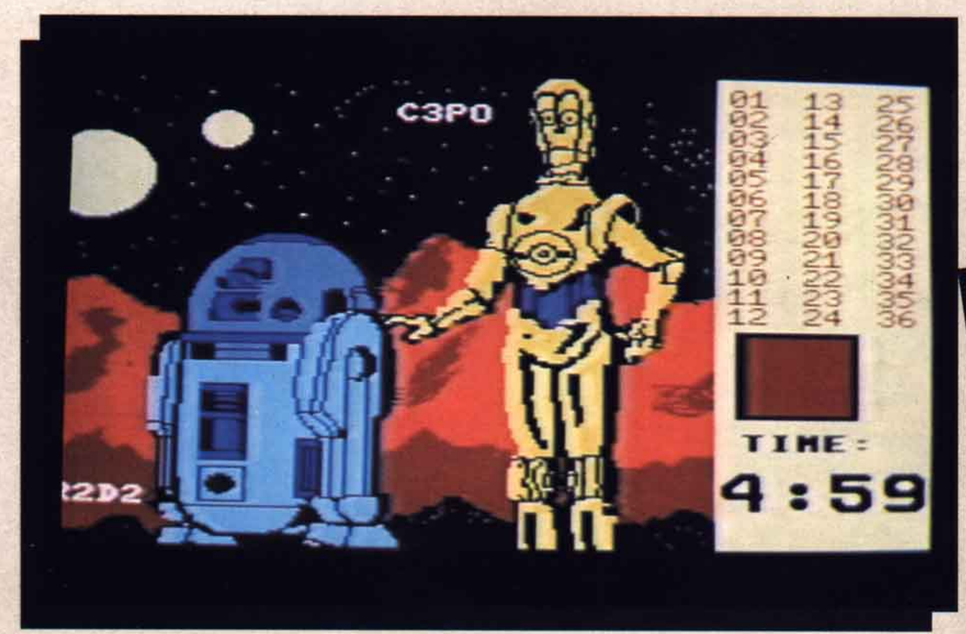


Estos dos son los primeros programas que la casa Walther Miller saca al mercado. Y aunque los nombres nos indiquen aventuras interplanetarias entre humanos y marcianoides, no es por este camino por donde se han encaminado los programadores de esta nueva casa de software. Sino que han buscado una fórmula nueva, pero no menos emocionante que el mejor de los matamarcianos. Se trata de construir el puzzle que hay en cada juego dentro de un límite de tiempo. Cada programa contiene en su memoria una pantalla gráfica que nos es mostrada durante unos instantes y seguidamente queda oculta bajo 36 cuadrados del mismo tamaño. Ahora nosotros debemos reconstruir la figura inicial, y para ello en el lado derecho de la pantalla hay una lista de 36 números que representan los 36 diferentes cuadrados a ordenar. Seleccionando uno de estos números se nos mostrará el trozo de dibujo que representa, y deberemos colocarlo en el sitio que le corresponde en la cuadrícula que hay en la pantalla. Repitiendo esta operación una y otra vez, debemos acabar por completar toda la cuadrícula, antes de que se acabe el tiempo de que disponemos.

Si lo logramos se nos felicitará y deberemos repetir la hazaña, pero esta vez en un tiempo menor. Dos de los mayores atractivos de estos programas es que los dibujos se basan en personajes muy conocidos por todos, los ewoks, R2-D2 y C-3PO, y que no siempre la misma figura corresponderá al mismo número, sino que esto varía en cada partida. Además si añadimos el fácil manejo del programa nos damos cuenta de que éste es un programa pensado y realizado para distraer a los más pequeños de casa, y para que empiecen a acercarse al ordenador y vean que es un instrumento útil y divertido.

Dos programas más, a sumar a la inmensa pléyade de juegos educativos para el público infantil. No solo de marcianos vive el MSX.

ANIMACION	*5*
INTERES	*6*
GRAFICOS	*7*
COLOR	*8*
SONIDO	*5*
TOTAL	***31***

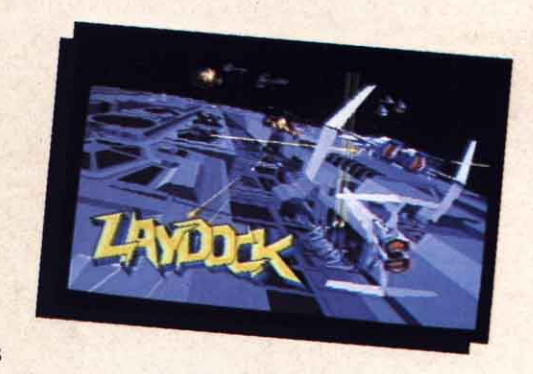




LAYDOCK

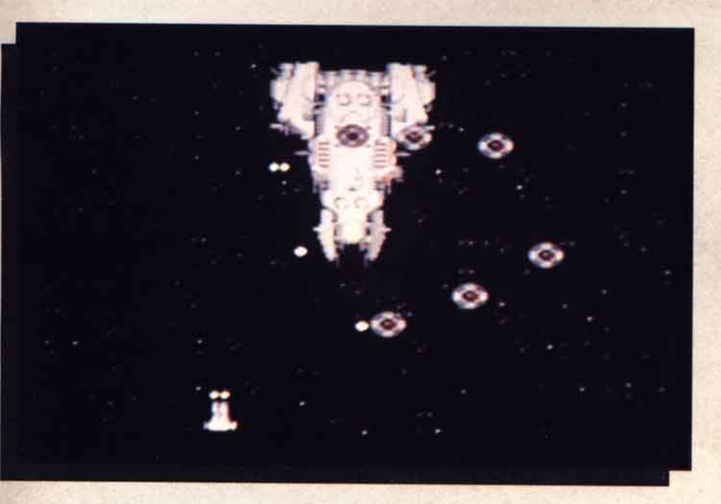
• TASOFT-SONY ■ JUEGO ▲ DISKETTE 3 1/2

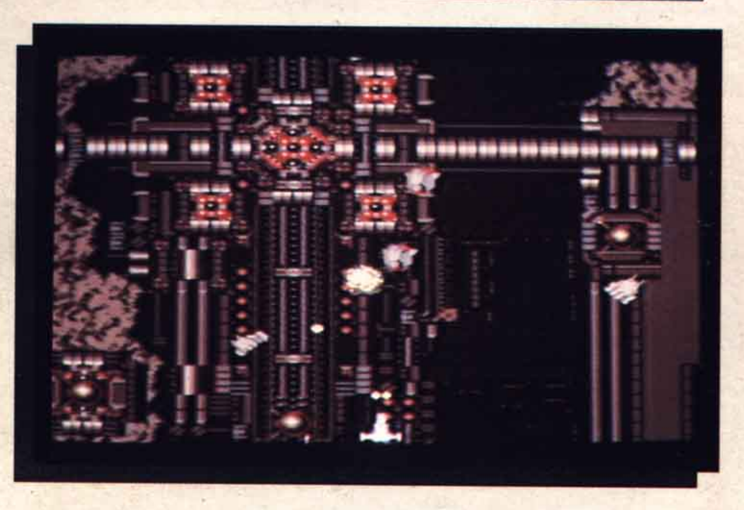
Son muchos los programas de matamarcianos que hay en el mercado, pero muy pocos los que tienen una calidad tan alta como la de LAYDOCK. Si hemos de buscar una razón a esta diferencia, encontramos la respuesta en los gráficos. Y todo porque LAYDOCK está pensado para los ordenadores de la segunda generación, y más concretamente para los ordenadores con lectora de diskette de doble cabezal. El programa se presenta en un formato de diskette 3 1/2, donde se encuentra toda la información acerca del juego, y continuamente el ordenador busca información en el diskette por cambio de escenario o por antiguas partidas que hayan sido salvadas anteriormente.
Para que veáis que no tan sólo es la calidad gráfica lo que hace de éste un programa original y sofisticado, a continuación os damos una lista de todo el armamento del que podemos llegar a disponer en nuestra nave:
TORA (Cañón de Rayos Láser): cuando se alcanza un cierto nivel, el Cañón Volcánico se convierte en el Cañón de Rayos Láser TORA, hecho que aumentará notablemente la capacidad de combate aire-aire del bombardero. Éste destruirá todas las naves enemigas que alcance. El poder destructivo de dicho cañón y su alcance se incrementarán a medida que aumente el nivel del jugador.
BULLPUP (misil aire-tierra): después del Cañón de Rayos Láser, la próxima arma disponible es el misil aire-tierra, conocido como BULLPUP. Ésta es la única arma aire-tierra disponible para los bombardeos en forma individual.
Las armas que podemos utilizar en opción de 2 jugadores, acopladas las dos naves, son las siguientes:
ADEN (Cañón Volcánico Multidireccional): ésta es la primera arma disponible después del acoplamiento de ambos bombarderos. El Cañón Volcánico Multidireccional ADEN puede disparar simultáneamente en cuatro direcciones.
BULLDOG (misil aire-tierra): el misil BULLDOG es una versión mejorada del misil aire-tierra BULLPUP. Su alcance es mayor, y su uso y control son de la responsabilidad del copiloto. Éste puede guiar este misil moviendo su palanca de control a izquierda o derecha para atacar objetivos terrestres, pero esto requiere muchísima experiencia.
CONDOR (misil aire-tierra): dichos misiles detectan la débil radiación electromagnética de las instalaciones militares del enemigo y dan en el blanco implacablemente. La precisión de estos misiles es superior al 95 %. Hay que tener en cuenta, sin embargo, que si el objetivo se encuentra demasiado cerca del bombardero, es decir, fuera de la mira de enfoque del misil, un sistema de seguridad autobloqueante impedirá su disparo.



KILLER: esta arma adicional es un campo de fuerza que rodea el Bombardero Tempestuoso y lo protege de los ataques del enemigo, destruyendo al mismo tiempo toda la nave que roce su esfera de acción.
ARMA SECRETA: los bombarderos disponen todavía de un arma opcional extra; ¡ya la descubriréis!
Pero si os explicásemos los miles de detalles que adornan el programa, este comentario sería inacabable, lo mejor es que conectéis vuestro ordenador, preparéis vuestro joystick y os pongáis a los mandos de la nave vengadora.

ANIMACION	9
INTERES	8
GRAFICOS	9
COLOR	9
SONIDO	8
TOTAL	**43**





WORLD GOLF

● SONY ■ JUEGO ▲ DISKETTE



Muchos son los programas basados en el deporte del golf, pero WORLD GOLF es el primero que está especialmente diseñado para los ordenadores de la segunda generación. La calidad gráfica queda bien reflejada en las fotos. La dificultad del juego es variable y para que podamos ir introduciéndonos en él de forma fácil y agradable se ha incorporado un modo de entrenamiento. En esta opción de juego nuestro objetivo es aprender a jugar con el ordenador, y experimentar con él, sin miedo a perder o no. Para realizar este entrenamiento podemos escoger en ir a jugar en cualquiera de los hoyos que están en el programa. Una vez tengamos suficiente confianza en nosotros mismos podemos ir al modo de competición. Aquí cada fallo será irreparable y sólo una mente fría y calculadora será capaz de terminar todo el recorrido por debajo del par del mismo.

Un aspecto muy importante de los simuladores de golf es de qué modo se nos permite hacer un lanzamiento, y de qué factores se hace depender a éste. En el WORLD GOLF los factores que determinan nuestros lanzamientos son los siguientes cuatro:

Selección de palos: Éstos se diferencian en cuatro grandes grupos: *wood, iron, wedge* y el *putter*. Los primeros son para lanzamientos largos, los segundos para tiros a distancias medias. El tercer grupo, formado por los palos PW y SW, son los encargados de sacar las pelotas de los bunkers. El último de los palos, el *putter*, es el utilizado una vez hemos llegado al green.

Dirección: este factor se representa gráficamente mediante una cruz que podemos hacer girar alrededor de la pelota, como si lo hiciera por encima de un círculo que tuviese su centro en la pelota. La dirección escogida es la de la recta, imaginaria, que pasa por la cruz y la pelota.

Potencia del golpe: éste es uno de los factores más innovadores del WORLD GOLF y determina a la vez la potencia del golpe y el toque de efecto hacia la derecha o la izquierda.

El cuarto y último factor es el impacto. Gracias a éste podemos realizar tiros largos o cortos, según lo que convenga en cada momento.

Una virtud de este juego es que hasta no haber efectuado el lanzamiento de la pelota podemos modificar cualquiera de los factores que acabamos de explicar.

Los factores que determinan la dificultad de un hoyo, y que nosotros no podemos controlar, pero sí que debemos tener muy en cuenta, son el viento, la inclinación del green y la posición del hoyo. Toda esta información está representada en la esquina superior derecha de la pantalla. Y además de dicha información al finalizar cada hoyo una gráfica nos mostrará nuestra puntuación hoyo por hoyo, y si estamos entre los mejores.



Seguro que si no nos distraemos con los gráficos, y tenemos un poco de constancia y paciencia, acabaremos por establecer algún nuevo récord.

Seremos los nuevos «Seves» de la temporada, aunque nuestros premios sean dólares de ficción.

ANIMACION	7
INTERES	8
GRAFICOS	8
COLOR	7
SONIDO	7
TOTAL	**37**

EL ZOCO

Intercambio todo tipo de juegos para MSX. Escribir a: Alex Marta. C/ Recó, 12. Castellar del Vallés. Barcelona.

Vendo SONY HB 75-80K, manuales, cables y 100 programas comerciales todo por 40.000 pts. Octavio Llop Salvado. C/ Córcega, 111, 1.° 4.ª D. 08029 Barcelona. Tel. (93) 230 16 92.

Cambio programas en disco de 3,5" para MSX-1 y MSX-2. Enviar lista a: Javier Leza. Aptdo de correos 420. 38080 Santa Cruz de Tenerife.

Intercambio programas MSX. Poseo entre otros: Green Beret, F. Martín, Dambusters, Road Fighter, Athletic Land, etc. Carlos Alvaro Caballero, C/ Francisco Salazar, 2, 4.ª. 39004 Santander.

Intercambio/vendo programas MSX (Green Beret, Zanac, Phantomas 2, F. Martín, etc.) más de 200 juegos. Manolo Martos Expósito. C/ Corredera San Fernando, 7, 3.° B. 23400 Ubeda. Jaén. Tel. (953) 75 15 32.

Intercambio programas (juegos, utilidades), para 1.ª y 2.ª generación del MSX con chicos/as de toda España . En cinta o disco 3,5". Prometo contestar. Jordi Dolcet Godia. C/ Dr. Huguet, 22. 25180. Alcarrás-Lérida. Tel. (973) 79 04 48.

Cambio juegos MSX. Tengo Knight Mare, The Goonies, Green Beret, Who Dares Wins II, etc. Javier Payarols. San Fructuoso, 51, 5, 1.ª. 08004 Barcelona. Tel. (93) 424 32 22.

Vendo cartuchos Green Beret, Antartic Adventure, Car Jambore por 6500 pts, estudiaré otras ofertas. También vendo otros 250 programas a 150 pts. c/u. Tel. (925) 23 26 84. Óscar López Pérez. Alberche 136-Edificio Granada. 45007 Toledo.

Intercambio programas de MSX en cinta, poseo todos los de Konami, Dynamic. Alfonso Moreno Gómez. Urb. Miraflores torre 7, 2.° 3.ª. Marbella 29600. Málaga. Tel. (952) 82 09 82.

Vendo programas MSX. Poseo más de 150 de los mejores. Entre ellos: Dunk Shot (Baloncesto), Twin Bee, Eggerland Mistery, Zanac, Gauntlet, Lucha libre, etc. También tengo más de diez copiones (cartucho, cinta y disco). Juan Luis. Tel (948) 23 63 85.

Vendo JUEGOS MSX, Pentagram y Pánico en las Vegas por 500 pts., también juegos de Konami como Twin Been, Knight Mare y Goonies. Programas de gestión y educativos Idea Base y Text, compilador Pascal. Tel. (93) 652 16 94. Antonio Montero. Sant Boi. Barcelona.

Vendo las siguientes cintas originales: Damas (Dimensión New), Base de Datos (master Computer), Traductor (Master Computer), U-Boot, Cartucho Road Fighter por 1.500, 1.200, 1.200, 500 y 3.000 pts. Vendo cinta con diez programas a escoger entre 250 programas comerciales por 2.200 pts. Jordi Arias Ruiz. Tel. (93) 654 48 22. Barcelona.

Vendo o intercambio juegos MSX de todas clases, poseo más de 100 títulos del mercado: Rambo, The Goonies, North Sea Helicopter, etc. Los vendo muy baratos. José Miguel Prieto López. Avd. General Castaño, 3. 06170 La Albuera. Badajoz. Tel. (924) 48 00 43.

Doy ocho juegos en cinta a cambio de un cartucho de Konami. Poseo los mejores de Erbe, varios de Konami: Taekwondo, Samantha Fox, etc. Llamar martes y jueves a: Rubén Fernández Santamarta. C/ Camarena, 82, 4.° A. 28047 Madrid. Tel. (91) 717 92 21.

Cambio toda clase de juegos para MSX. Poseo más de 40 títulos (Hero, Goonies, Batman, etc.) Carles Baltrins. C/ Detrás de la Iglesia, 1. 17300 Blanes. Gerona. Tel. (972) 33 13 13.

Compraría ampliación de memoria de 64K así como cartuchos de juegos. Jorge García. Plz/ de los Mil Delegados, 1. 3.° A. 28041 Madrid. Tel. (91) 211 65 67.

Vendo ordenador Philips VG-8020 (80K), por cambio de sistema. Carlos Cabañero Pérez. C/ Brescia, 15, 3.° 7.ª. Madrid. Tel. (91) 245 24 53.

Compro toda clase de programas comerciales para MSX-2. Mandar lista con precios a: Javier Pérez. C/ M.ª de Fitero, 34, entlo.-B. 31011 Pamplona. Navarra. Tel. (948) 26 83 88.

Intercambio juegos de MSX, poseo Karate Internacional, F. Martín, Donkey Kong, Yie are Kung-Fu I, Jack the Nipper, etc. Juan Manuel García Gómez. Los Rosales portón 31, 1.° B. Ceuta. Tel. (956) 51 55 03.

Vendo/cambio programas de MSX-2 poseo: Red Light of Amsterdam, Chopper II, Badmax, Laydock, Perry Mason, Nemesis, etc. c/u 2.000 pts. Antonio Muñoz Rando. C/ Energía bl/m esc/1 10 2.°. Barcelona 08004. Tel. 332 73 34.

Cambio juegos MSX (Goonies, Green Beret, Avenger, Batman, Kung-fu M, etc.) Deseo Némesis, Twinbee, Crusader o copiones de cualquier tipo. Estoy interesado en copiones Turbo. Manuel Angel Pajuelo. C/ Teatro, 15. Peñarroya. Pueblonuevo. Córdoba.

Contacto con usuarios de MSX-2 y MSX-1 con unidad de disco, para intercambiar información, trucos, etc. M. E. Martínez. Alfonso I, 28 6-B. 50003 Zaragoza.

Intercambio programas MSX. Juan Sánchez Sevilla. C/ Guzmán el Bueno, 18, 2.°. Tarifa. Cádiz.

Intercambio juegos, poseo entre otros F. Martín Basket Master y 30 más. Carlos Asensio Estevez. C/ Almazán, 27, 1.° D. 28011 Madrid. Tel. (91) 463 78 50.

Vendo programas de MSX, 20 juegos de los últimos, The goonies, Green Beret, y otros por 300 pts. Me interesaría programas de morse y relacionados con la radioafición. Juan José Ritoré, apdo. 20039. 06080 Badajoz.

Cambio/vendo/compro programas desensambladores, ensambladores, copiones, procesadores de textos. Solamente con chicos/as de mi provincia. Pedro J. García. C/ Miguel Jiménez Rueda, 22. 18230. Atarfe. Granada.

Compramos/vendemos juegos más de 200. Primeros títulos. Todos los de Konami, incluso los que no han salido en España. Return Club MSX. C./ Soto del Rey, 24. 28350 Cienpozuelos. Madrid. Tel. (91) 893 00 47.

Intercambio programas; somos un grupo de chicos que tenemos dos ordenadores MSX y Spectrum 48K. Jorge Martínez Zoreda. C/ Almazán, 27, 2.° C. 28011 Madrid. Tel. (91) 463 47 37.

Cambio juegos MSX, poseo más de 100 títulos: Dunk Shot, Goonies, Avenger y las Vegas. Me interesaría principalmente Livingstone S y Némesis. Raúl Arija. Avd. Valencia, 5, 1.° B. Burgos. Tel. 26 62 85.

Intercambiamos todo tipo de programas MSX con España y el extranjero. Tel. (956) 60 47 81. Círculo Informático Algecireño. C/ Granero, 1-2 izq. 11206 Algeciras.

Vendo cartucho MSX Green Beret por 2.500 pts. José Luis. Tel. (91) 217 62 86. A partir de las 7.

Vendo ordenador Toshiba HX-10, MSX de 80K completamente nuevo, con cables, libros, cinta demostración, juegos y un cartucho de TENNIS KONAMI. Todo por 25.000 pts. Jorge Beltrán. Tel. (96) 178 39 64. Valencia.

EL ZOCO

Cambio/vendo programas MSX. Poseo más de 330 programas como: Arkanoid, Twinbee, Phantomas II (últimas novedades). Mandar lista. José L. Rus. C/ Colón, 27, 2.° A. Granollers. Barcelona. Tel. (93) 870 97 75.

Cambio programas de todo tipo, me interesa sobre todo el Spitfire 40 original ofrezco 20 juegos por el programa, sólo si es original. Mandar lista. José Manuel Benítez Vega. urb. Maripins, chalet 16. Salou. Tarragona. 43480. Tel. (977) 37 13 39.

Vendo ordenador HB-75P de Sony en perfecto estado por 45.000 pts. Regalo cartucho de ampliación de memoria DATA Cartridge de 4K Bytes y manuales de funcionamiento. Mariano Juan González Martínez. C/ Dr. Profesor Severo Ochoa, 12, 4.° 9.ª. Valencia 46010. Tel. (96) 360 06 46.

Intercambio programas MSX, en cinta o disco 3,5" poseo más de 600 títulos, últimas novedades en el mercado español y extranjero. Mandar lista. José Hernández Sánchez. Avd. de la Coruña, portal C, 2.° A. Noya. La Coruña.

Vendo juegos MSX poseo muchos títulos. Si puede ser de Barcelona. Javi. Tel. 332 54 29. Barcelona.

Intercambio los juegos Astronomía, Maths, Contabilidad, Super Sky, Pokes para juegos (listados), por 400 pts. o por algún juego no original de estos: Night Mare o Commando. Antoni. Tel. (973) 47 00 84.

Vendo ordenador PHILIPS MSX VG8020, joystick, 5 juegos de cartucho. Todo por 55.000 pts. Llamar de 9 a 11. Tel. (96) 242 04 19. Vte. José López. C/ Echegaray, 4. 1.° Algemesi. Valencia.

Compro ampliación de memoria 64K de cualquier marca. También compro unidad de disco. José Luis Stoduto García. C/ Manuel Blasco, 8, 5.° C. 42004 Soria.

Cambio juegos MSX (tengo casi todos los de Konami y 130 más). Busco ensamblador a ser posible el Gen Mon. Me interesan: Némesis, Jail Break, Basket Ball, Rambo, Commando, etc. Alfonso Montiel o Miguel Angel. Tel. (968) 46 99 02.

Vendo ordenador TOSHIBA HX-10 64K, Joystick Canon, 100 juegos, 20 revistas por 30.000 pts. Monitor PHILIPS color CM8500 por 39.000 pts. Unidad de disco 3,5" Sony HBD-50, 10 discos por 55.000 pts. Todo por 120.000. Bruno Lombarte. C/ Miranda, 77-79. Cornellá de Llobregat. Barcelona. Tel. 376 17 14.

Cambio/vendo juegos MSX. Poseo más de 200. Entre ellos: Super Strip Sex, Copy Guay V. 3, Phantomas II. También poseo pokes y trucos para los juegos y en Basic MSX. Me gustaría saber información sobre manejo y práctica en las impresoras MSX. Javier Rosendo López. Avd. Concha Espina s/n.° Cabezón de la Sal. 39500 Cantabria. Tel. (942) 70 05 16.

Intercambiamos todo tipo de programas MSX con España y el extranjero Círculo Informático Algecireño. C/ Granero, 1-2, izq. 11206 Algeciras. Tel. (956) 60 47 81.

Vendo ordenador SONY HB-55P, ampliación a 32K, TV Philips B/N, Plotter, Sony prn-c41 por 50.000 pts. Matías Zaya. Tel. 44 54 59, de 6 a 9 de la tarde. Málaga.

Intercambio juegos de MSX. Mis mejores juegos son: Las tres Luces de G, Futbol Soccer, Ciberun, Boxeo, F. M. Basket, Knight Mare y Kung-Fu Master, etc. Me interesan Green Beret y The Goonies. Pablo Gómez Prieto. C/ Legión Española, 9, 6.° A. Huelva. Tel. 26 16 98.

Cambio los siguientes juegos: Nightshade, Speed King y Space Walk por un copión que pase de cartucho a cinta. Rene Mueña. Urb. Tres Carabelas. Edificio La Pinta P 1.° B. Las Palmas de G.C. 35016.

Vendo ordenador Sony HB-101P con embalaje original, expansión memoria HBM-64, cassette Panasonic RQ-8100, instrucciones, cables, 2 libros, revistas, màs de 50 juegos, joystick Quick Shot II MSX por 55.000. Negociables. Manuel Romeu. Tortosa. Tarragona. Tel. (977) 44 24 59 de 2,30 a 5,30 tarde.

Intercambio juegos con chicos/as de toda España poseo entre otros Samantha Fox, Pippols, Kung-Fu Masters, hasta 140. Carlos Fdez. Díaz. Poblado C.T. de Escombreras edif. B 1.° izq. Valle Escombreras. Cartagena. Murcia 30350. Tel. (968) 55 22 87.

Intercambiamos/vendemos programas en cinta y disco de 3,5". Más de 400 títulos de importación y nacionales. Club Drakes. Manda tu lista. Apdo. 5083 de Madrid.

Cambio programas MSX poseo más 250 en código máquina. Paco Salazar. C/ Pagoeta Bidea, 4-5. Erandio. Vizcaya.

Vendo veinte estupendos juegos de KONAMI (Knightmare, Sky jaguar, Goonies, Time Pilot... etc) por sólo 3.950 pts. Garantía de que los programas funcionan todos bien. Martín López Corredoira. Polígono del Ceao, 71. Lugo.

Vendo video juegos Philips G7200 con monitor incorporado, sin usar, con dos mandos para juegos, más de 3 cartuchos por 25.000 pts. Albert. Barcelona. Tel. (93) 321 11 72. Precio discutible.

Intercambio programas MSX. Poseo Konami's pasados a cinta y novedades como: Livingstone, Cosa nostra, etc. Estoy interesado en el copión Tonus y el juego de Konami: Némesis. Tengo más de 120 juegos y utilidades. Pere Franquesa. C/ José M.ª Pellicer, 47, 1.° 1.ª. Ripoll 17500. Gerona. Tel. (972) 70 14 95.

Vendo ordenador Dragon 200, un monitor color, 15 juegos, precio a convenir. Mercedes. Tel. (93) 230 40 57.

Club MSX para intercambio de programas, ideas y trucos. J. Michel Jarre. C/ Santa Agata, 9, 3.° 2.ª. 08012 Barcelona. Tel. (93) 218 90 23.

Vendo MSX Sony HB-F-500-P con unida de disco de 1 mega, teclado independiente y numérico, 128K de URAM, conexión para monitor. Regalo sistema operativo MSX-DOS y muchos juegos comerciales. 100.000 pts. Luis Segura Salvador. Plaza Castellini, 1, 3.° B. Cartagena - Murcia.

Intercambio programas MSX, poseo todos los mejores títulos. Iván García. C/ Utrillas, 12, P 3.° B. 28043 Madrid. Tel. (91) 759 23 35.

Vendo impresora PHILIPS VW0020 de 80 columnas, poco uso, por 30.000 y además regalo procesador de textos y base de datos. También vendo todo tipo de juegos MSX en cassette o disco de 3,5'' por 500 pts. Llamar al Tel. (93) 330 81 31.

Cambio juegos comerciales MSX como, HERO, SOCCER, FIRE FOX, etc. Tengo más de 100 juegos. Manuel Domingo Murillo. C/ Salvador Ricart, portal, 30; pta. 11. 46470 Albal - Valencia.

Pensionista 53 años, agradecería donación de curso de electrónica o sonido con materiales AFHA, CCC, etc.; como terapia ocupacional. Envío relación de revistas y libros para posible canje. Julián Seguen. C/ Serradilla, 28. 28044 Madrid. Tel. (91) 705 98 24.

Vendo PHILIPS VG-8020 de 80K, muy nuevo y poco uso, con manuales en español y cables. Además juegos primeras marcas. Precio a convenir. Javier Moro. Tel. (93) 386 62 86. Sta. Coloma de Gramanet - Barcelona.

Esta Primavera Sony da mucho juego

NUEVOS PROGRAMAS MSX



LAS VEGAS. Juego de detectives. Deberás recuperar una alta suma de dinero de un «capo mafioso».

ROMA. LA CONQUISTA DEL IMPERIO. La conquista de las tierras del imperio romano.

LODE RUNNER II. Excitante juego de aventuras y prueba de habilidad.

SPACE KIT. Para dibujar el espacio como tú siempre lo has imaginado.

PRINT LAB. Diseñador gráfico. Incluye un cassette con 19 interesantes trazados y dibujos.

GRAPHIC MASTER. Editor gráfico.

MUSIC STUDIO. Para componer canciones sin necesidad de tener conocimientos de música.

FARM KIT. Construye y pinta la granja de tus sueños.

NUEVOS PROGRAMAS MSX2



CHOPPER 2. Al mando de un helicóptero blindado deberás combatir a tus enemigos.

RED LIGHTS OF AMSTERDAM. La más excitante partida de poker que jamás hayas jugado.

HYDLIDE. Programa de acción. El príncipe debe rescatar a la princesa con muchas dificultades.

WORLD GOLF. Juega al golf como un profesional.

LAYDOCK. Eres el piloto escogido para devolver la paz al Universo.

KINETIC CONNECTION. Forma la figura misteriosa atrapando los objetos voladores.

¡No te los pierdas!

